DOCUMENT RESUME

ED 150 977 IR 005 661

TITLE Proceedings of Industry Workshop on Information Systems Education.
INSTITUTION Data Processing Management Association, Park Ridge, Ill.
PUB DATE 76
NOTE 71p.

EDRS PRICE MF-$0.83 HC-$3.50 Plus Postage.
DESCRIPTORS Computers; *Computer Science Education; *Educational Programs; Higher Education; Industry; *Information Systems; *School Industry Relationship; *State of the Art Reviews; *Workshops

ABSTRACT

Presentations delivered at the Industry Workshop on Information Systems Education are included in this report. Three topic areas are covered: (1) state of the art reports; (2) what industry thinks it needs; and (3) opportunities for industry/academic interaction. These are followed by texts of the reports of workshop working group leaders and resulting discussion. A report on the American Association of Collegiate Schools of Business, and a final session reviewing recommendations made by the working groups conclude the document. A list of conference participants and the Board of Regents of the Data Processing Management Association Education Foundation, as well as a description of its research, programs, and services are appended. (KP)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED150977

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

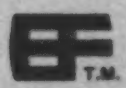
Proceedings of

Industry Workshop on Information Systems Education

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Roland D. Spaniol

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."



Data Processing Management Association
Education Foundation

IR005661

Copyright © 1976 by the Data Processing Management Association, 505 Busse Highway, Park Ridge, IL 60068. All rights reserved. No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of DPMA.

The DPMA Education Foundation was incorporated in 1975 in the State of Illinois as a not-for-profit corporation. Operating under a broad and ambitious charter as a charitable/educational institution, the Foundation is dedicated to expanding educational opportunities for information systems professionals. Contributions to the Foundation are tax deductible. The Foundation's objectives are spelled out in detail in the Appendix.

Table of Contents

Introduction	2
Section One: State of the Art	3
Gopal Kapur	4
Gene Smith	8
James McKenney	10
Section Two: What Industry Thinks It Needs	15
Mayford Roark	16
Robert Ott	21
Harlan Mills	25
Section Three: Opportunities for Industry/Academic Interaction	30
Frederick Withington	31
Section Four: Workshop Discussion Groups	41
Max Hopper	42
Paul Strassmann	44
D.R. Wolf	47
Felix Kaufman	49
Richard Canning	52
Section Five: American Association of Collegiate Schools of Business	54
Dr. B. L. Barnes	55
Conclusion	58
Appendix	63
List of attendees	63
Education Foundation	64

2

Introduction

On April 6, 1976, the DPMA Education Foundation sponsored an Industry Workshop on Information Systems Education. A distinguished group of invited attendees spent the day discussing ways to improve educational opportunities for those entering industry and government as information system designers.

Profile of attendees

The attendees were senior managers from industry and government; consultants, publishers and public accountants; and representatives of educational institutions that offer programs which we believed noteworthy. Our guests included the presidents of AFIPS and DPMA, and a representative of the American Assembly of Collegiate Schools of Business. A List of Attendees is included in the Appendix.

The Workshop began with three presentations: First, a session on the state-of-the-art in information systems education; second, a discussion by several industry spokesmen of the skills and personal characteristics they seek when recruiting college graduates and those they attempt to develop subsequently in training programs; and third, a session intended to suggest specific actions that industry and academia might take to meet industry's needs while satisfying academia's requirements for scholastic content and excellence.

These presentations were followed by working group discussions and reports, and a wrap-up session in which the recommendations made by the various working groups were discussed.

Three firms helped make the Workshop possible: Bank Virginia Service Company, IBM Corporation and Reynolds Metals Corporation. We are most grateful for their generosity. And we are most appreciative of the stalwart efforts of Dr. Howard B. Wilson, General Director, Management Information Systems for Reynolds and a former Regent of the Foundation, who managed our fund raising efforts.

What follows includes papers presented at the Workshop and an edited summary of the subsequent recommendations and discussions. We hope that you will find both stimulating and that you will identify opportunities therein to join with us in our efforts to improve the quality of educational programs for information systems professionals.

George Glaser, General Chairman
Gene Smith, CDP,
Program Chairman

Section One: State of the Art

The opening panel session was on State of the Art, and was led by Gopal Kapur, author, lecturer and consultant in the field of information systems management and education and president of Kapur & Associates, Danville, CA.

Other participants on this panel included Dr. Eugene Smith, CDP, vice President of the Education Foundation's Board of Regents and associate professor, Department of Analysis & Research, Texas A&M University, College Station, TX, and Professor James McKenney, Harvard University Graduate School of Business, Boston, MA.

Questions which were asked after the presentations are included at the end of this section.

4

Gopal Kapur

I would like to thank the DPMA Education Foundation for inviting me to chair this panel.

First, I will explore the problems of continuing education as well as present some examples of innovative programs currently being offered at some colleges and universities. Professor Smith will then review the philosophy behind the curricula at Texas A&M University; Professor McKenney will discuss the curricula recently implemented at Harvard University.

It is very satisfying to note that the emphasis in this investigative seminar has shifted from "computer science" education to "information systems" education. I certainly hope that we can refrain from discussing the number of mathematics and electrical engineering courses needed to give information systems curricula the note of respectability. The keynote addresses delivered by George Glaser at the 1974 NCC and 1974 EDUCOM in Canada served to heighten this interest and controversy in computer science and information systems education.

The courses presently being offered in colleges and universities have been both attacked and defended vigorously: The main criticism being that educational institutions do not prepare their graduates for real-life situations. Coupled with this dilemma of inadequately prepared new graduates is the problem of an abundance of poorly trained, underqualified and inefficient existing data processors. It should be obvious that the full potential of computers cannot be realized until these problems are seriously addressed and definite actions taken to improve the sinking "state of the ark."

Institutions behind

Educational institutions are usually four to five years behind the state of the ark because DP is one of the few areas of education and development which is largely initiated, researched and developed by vendor organizations and not by colleges and universities.

Consider, for example, data base systems, interactive systems development and operations, telecommunications and distributed networks, to name a few. Very few educational institutions have respectable courses to educate people towards preparing them for user industry jobs in these areas. Many DP managers ask, often quite bitterly, "What are universities up to? Why are they not preparing people for today's needs?" In fact, it is almost impossible to find new college graduates who can help define, assess and implement data base systems—a critical and growing need for large computer users.

Another example is the recent proliferation of virtual operating systems within the business data processing industry. The DP practitioners' first introduction to these systems often is on the day they are installed and then they are supposed to use them efficiently and cost effectively. Practically, no education is available in this area from educational institutions.

Structured design and structured programming for business data processing were not developed through the college/university propaganda. Colleges and universities have been negligent in researching the entire area of

programmer productivity. Dr. Dijkstra wrote his now famous paper, "Go To Statements Considered Harmful," in early 1968 and the book *Structured Programming* in 1972 (co-authored by Dahland Hoare). But to this day there are practically no respectable courses available in structured methodology and program engineering in educational institutions. The truth of the matter is that most college/university teachers have had little, if any, exposure to these system design and program development methods.

Instant potato

DP managers who believe that a maximum of a three-day quick course is all that is needed to train and educate DP practitioners have what I'd like to call the "instand potato" syndrome towards systems analysis and programming education and training.

This is very evident from the large number of short seminars offered in such areas as structured design and programming, advanced COBOL programming, systems analysis and design, to name a few. Of course, programmers and analysts can't be spared for more than three days at a time. What would happen to all the "bugs" which come out of the programs when they are away?

It is interesting to note that neither industry nor educational institutions have been able to agree on the basic definition of information systems education; and worse yet, there is little meeting of minds on their common needs and objectives. During a recent one-day seminar at the University of California at Irvine, titled "Software Engineering Education: Needs and Objectives," no specific definition or meeting of minds emerged.

Here are excerpts from some of the papers presented: "...coding techniques are not taught. Some of the things that the working programmer spends a great deal of his working life doing—coding, debugging, documentation—are considered perhaps, too trival (or maybe unexciting) to dwell on in school ... Programming is not yet the trivial matter much of the academic community would like to think it is. Programming is a legitimate part of the profession." (D. E. Anderson, McDonnell Douglas Automation Co.)

"We need people who are able to integrate software and hardware systems, often with various pieces coming from d i v e r s e sources. Furthermore, we need people who can *design* systems and programs. Graduate education in the field is less important than continuing education, which should be made widely available ... Universities have trained a large number of architects, but few of them ever get to design a Seagram building; most draw the plans for tract homes ... I hope those of you who will design university software engineering programs will keep in mind the "tract home" needs of the commercial data processors." (Robert H. Clinton, Bank of America)

"We generally bring in about 25-50 people per year from universities. We find in general that they aren't very well educated to perform several software engineering functions which are particularly important in our work. Some of the topics of interest noted in this paper included 'expressing oneself in clear English, developing software requirements and design specifications, developing applications software, performing economic analysis, working with project management techniques and working in groups." (Barry W. Boehm, TRW Systems)

These are strong indictments of an educational process supposedly preparing graduates to enter the job market. It is my opinion that the educational community has failed to realize the importance of disciplined and practical approaches to information systems education. Proof of this is the lack of adequate, much less excellent, courses in colleges and universities throughout the country.

Industry needs

The industry needs people who are trained in the real-life situations rather than in nonrealistic approaches to education. The California State College System (the state university system) has over $3 million in computer systems but only two of the campuses offer any education related to business data processing at undergraduate levels. Graduates from the ones which make such offerings are in much demand. So the industry needs are there. To my knowledge none of the University of California campuses offer degree programs in business data processing. The very use of the word COBOL in the hallowed halls of certain computer science departments is considered an act of heresy.

I still marvel at the often-advertised course "FORTRAN for Business Applications."

Other subject areas in which students need education and training are hardware/software evaluation and selection, system and program maintenance, hu-

man engineering, communication skills and management skills. Hardware evaluation and assessment are of vital interest to the user community. Users are continually bombarded with proposals for new hardware by vendor salesmen. Most DP departments are not really well equipped to evaluate such proposals and usually are at the mercy of the suppliers. There is very little education and training in this area being imparted anywhere.

Software

The cost of developing and using software is high and continually rising. Most programmers spend a majority of their time developing code, but very little education has been, or is being, imparted in systems analysis, software development and implementation. To most college students, software maintenance is an abstraction, while it is the day-to-day reality of a programmer's life. College students are seldom required to maintain existing pieces of software. In fact, I know of undergraduates who have never written a program over 50 lines of source code. We need to give computer professionals education in the area of psychology of people to better understand the man-machine interface.

One often hears the remark, "If it weren't for the human beings we'd have little problem with computer systems." What a poor commentary on the people responsible for computer systems. The human interface area has been continually neglected in information systems education.

Communication skills (oral and written) also have been neglected for a long time. Unfortunately, very few business systems analysts develop communication skills; as a result we find over and over again that what the user wanted is not what he gets. Why is that? To a large extent, it is because of a lack of communication between the users and DP.

Four generations

Computer hardware/software technology is dynamically growing and changing. It can be safely said that during the past 12 years we have experienced four generations of computers. This requires a certain flexibility among DP practitioners. If we train a graduate who is a specialist today, there are good chances of him/her becoming obsolete in a very short time.

Today's specialist may not be able to adapt to the newly developing technology (hardware and software).

The blame

Industry is not above blame for the current lack of "good" information systems education. Historically, industry has been too short-sighted in the area of professional training and continuous skills upgrading. Education and training budgets are pitifully meager and are the first to be slashed in case of any type of budget crunch. Of course, it takes years before such budgets can be restored to adequate levels again. The surveys, undertaken by *Datamation* for 1974 and 1975 regarding corporate DP budgets showed that the cost of personnel has crossed the 50 percent mark. The real shocker was the DP training and education figures. Close to .68 percent of the DP budget was allocated, on the average, for education and training.

It is not uncommon to find DP departments with an annual budget of well over one million dollars and the education budget of no more than $10,000—a meager amount for education of the people who are so inadequately prepared for their jobs in the first place. Perhaps that is why "productivity of programmers is too low, the overall systems utilization is poor, and software quality is substandard."

Osmosis

Some new software systems are said to be transparent to the data processing staff ("foggy" would be a more appropriate word to describe the situation).

I know of companies which have either reduced or not enhanced their DP budget because the vendor salesmen assured them that new data base systems and operating systems could be implemented without a noticeable ripple. Such systems are alledged to be automatic and, of course, the chief selling point revolves around more automation and less human intervention. There is only one trouble with these new "automatic" systems. The vendor representatives forget to mention where the "solution button" is located. In most cases the user industry ends up spending twice as much money and resulting systems are anything but satisfactory.

No initiative

The lack of concern for DP education and skills upgrading displayed by so many DP managers/directors is mindboggling. During many of my addresses to DPMA and ASM chapters, I have often asked the questions, "How many have their DP training/education needs identified? How many have contacted local col-

leges and universities asking for (or demanding) curricula in skills upgrading? How many would sponsor (in time and money) such courses?" The show of hands is seldom encouraging.

Exchange program

We need to set up an industry/academic exchange program. But for too long the academic world has been looking for "degreed" people to satisfy "their accreditation requirements." It is really not possible to find people who have the required advance (or terminal) degrees who have experience in industry and who are willing to go into full-time teaching. I think certain avenues have to be found where one can temporarily bypass the politics, the bureaucracy (and at times the sheer snobbishness) of the academic environment to attract qualified industry people into colleges and universities.

Unfortunately, teaching a course once a week in night school (often located at a local high school or junior college), does not bring industry people into the mainstream of the academic world where they would have the opportunity to participate in meetings, policy-making bodies and long range planning. Instead we must create new jobs (hopefully fully subscribed by the industry) for visiting industry teachers on a revolving basis. I believe one to two years tenure would be most beneficial. We can't overlook the complicated logistics for allowing (even assuring) these people's return to the industry.

On the other side, industry needs to establish positions for visiting faculty and advance degree students. Again, their problems are not to be thought of as short summer assignments. Such short tenures do not give the faculty enough time to get involved in the total information systems development picture.

Industry should encourage appointments ranging from one to two years which would allow visiting teachers regular staff/-managerial assignments. Some of the position titles could be: Programmer/analysts, system analysts/designers, project managers, assistant to the DP manager and internal consultants. This would expose the academicians to the real world of DP and prepare them for imparting "useful" education to their students. Unfortunately, not much has been done by industry in this critical area.

The problem is further compounded by the fact that most faculty members would rather spend their time in theoretical research. Practical experience of the type outlined above has not gained respectability within the colleges and universities.

Two examples

I would like to finish my discussion at this point by citing two examples of what I have been talking about. Recently I received information from Babson College in Massachusetts regarding their academic/industry program at continuing education in DP. This program has been established in association with 10 corporations that provide most of the facilities and money to support a series of courses on such areas as: Systems analysis, systems design workshops, DP management methods, data base concepts and design, on-line systems, programming methods and strategies. The program was started two years ago and nearly 400 participants from the industry have already taken advantage of the offerings.

A similar approach has been taken by the Center for Computer Education within a New School for Social Research in New York (I am not personally familiar with these two programs and as such am unable to remark on their quality).

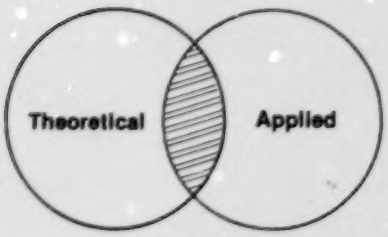
Summary

It is my opinion that the educational community has failed to realize the importance of disciplined and practical approaches to information systems education. Proof of this is the lack of adequate, much less excellent, courses in colleges and universities throughout the country. At the same time, industry has abdicated its responsibility in the area of support of continuing education and training.

In this session the panel will present their experiences and thoughts on the development of "realistic" information systems curricula to meet both the short range and long range needs of industry and explore the means of bringing existing data processors to a high level of competence. The underlying theme will be the development of "joint industry/academic programs" in bringing about the much needed change in information systems education.

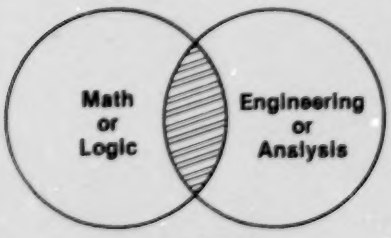
Gene Smith

I'd like to take just a moment to introduce some controversial issues in terms of education. If we think in terms of a Venn diagram, I can visually identify some problem areas. Here are the kind of things which to me are controversial. For one thing, there's the approach. On one side we might have a theoretical approach. On the other side we might have an applied or practical approach.



Approach Factors

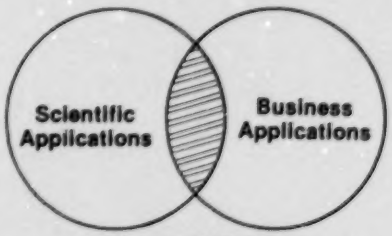
It seems to me that this is a very controversial issue among academicians. Another area might be an approach in terms of math or logic on one side and an approach in terms of engineering or analysis on the other side.



Approach Factors

Which approach do you take in your training?

And still another factor might be: Do we train our data processing, or computer science, or information students in terms of scientific or engineering applications, or in terms of business applications?



Approach Factors

It seems to me that each university you visit has one very definite approach and the students they turn out will be oriented towards that approach. In terms of training, what kind of training are we talking about?

- Continuing education
- Technical training
- Associate degrees
- Bachelor's degrees
- Master's degrees
- Doctoral degrees

First let's consider continuing education. At Texas A&M we have a very vigorous continuing education program. When you look at the emerging two-year colleges and the technical schools, you might have such things as an associate degree. And then another somewhat controversial item—should we turn out a student with a bachelor's degree in computer science. Believe me, we could get into a very hot argument about that topic. I don't think it is worth discussing at this time, but it is one of the controversial issues. Then we have the Master's and Ph.D. degrees above this. After having identified the factors, I'd like to tell you something about

Texas A&M's approach to information systems or data processing and computer science education.

Like so many other universities, we have one camp that is engineering oriented. The first computer courses were taught on the engineering side of the house. We started out in about 1962 with an M.S. in computer-science and what we call an M.C.S. (Master of Computing Science), which does not require a thesis. We then added a doctoral program, and about two years ago a B.S. in computer science was added.

On the business side, we began offering an M.B.A. in business computer science in about 1967 and then a Ph.D. in business with an area in business computer science. We weren't the first but we were early to recognize that we needed to integrate the computer into our education processes. At the undergraduate level in the College of Business, we don't try to offer a degree; however, we do have required courses for all business students. We have optional courses if students are interested and have enough elective hours to take them.

Content view

Although my degrees are in engineering, it seemed to me that the real challenges lie in the business area and so I was converted some time ago. I have been teaching for some time in the College of Business. In the M.B.A., of course, we offer the traditional common b o d y of knowledge courses and had we not done this we would not be accredited. So we have pressure as to how much DP work we can teach from the accrediting agency, in this case the AACSB.

In the computer area, we require a certain level of competence in that a student should not g r a d u a t e and then call himself a computer scientist or data processor unless he has a knowledge of a variety of languages. At our university, the languages are taught in the College of Engineering. Now I have some irritation about how they are taught but I think this is a good place to teach them. We offer a variety of optional courses for the M.B.A. student including an MIS course, a data base concepts course, a data processing management course and a systems analysis course. In an attempt to broaden the student, we require that he take either an internship (which involves a significant full-time work assignment followed by a paper on his experiences) or maybe a problems course where he does some specific problem which includes a formal paper.

I would like to identify some problems that we and other universities have. One of the problems is that change is difficult. That is, if we see a new trend coming, we can't convince our colleagues and the academic council or the coordinating board that a change should be made in a very short period of time. Suppose that we feel that a new course should be taught next semester. It's a fairly long drawn out process to make a significant change in direction. We have both the internal and external factors that slow us down, impede us and perhaps help us so we're not impulsive.

Funds

One of the most significant problems is that we have a limited amount of funds. In our university, in our college, we don't use terminals and I know a lot of smaller colleges that are doing a better job of teaching because they can get their students on the terminals. We have a super computer. We have an Amdahl, but our internal DP budget has not been increased in two years although the number of students has increased.

We're currently one of the fastest growing universities in the United States. We're still building dormitories. And I think one of the most significant problems which, hopefully, we'll face in some part today is that it is very difficult to keep the academic type current. You all know how fast everything is changing. I've been at Texas A&M for nine years now and I'm taking a year's leave to go to Washington in an attempt to upgrade my knowledge of data processing techniques. This leave is under the auspices of the AACSB, which has an on-going fellowship program in cooperation with schools of business.

I'd like to leave you with these problems: Change is difficult, funds are limited, and we need to keep current.

10

James McKenney

Last night someone mentioned a very interesting fact—very few of us are retired. In fact, we're probably the first generation of human beings that has ever lived through four generations of something and they're looking forward to redefining that problem. I agree with Kapur that industry in general is ahead of university computer teaching programs. In this meeting, we have the charge of attempting to provide suggestions for bringing the two together. I suggest that our opportunities are to try and merge what we know in industry with communicable skills and research skills in the university. I'm concerned also that the real problem is the heterogeneity.

Lagging behind

There are good universities and bad universities and some good programs; but in computers, in general, we're lagging behind the best in the industry. However, there are some opportunities for improving industry practice. One of the opportunities I'm currently involved with is the U.S. Treasury; and we're making an audit of how they use computers. Amusingly enough, last winter we stumbled across an IBM 609. Remember that device? The Bureau of Engraving and Printing did their payrolls on a 609 with several professionals—called system managers—feeding cards into it.

They were pleased to have zero hardware costs. The line item for hardware was nothing. Yet, unfortunately, the line item for systems managers was extraordinary. There were several other omissions and I am sure we can all bring out skeletons. That is not the issue.

I think our academic problem is to inculcate students in the art of the possible. That is, what can the technology do to help in management and the operations systems today? How to assess organizational realities and develop the best feasible system for a given situation.

Theory

I think we need a certain amount of theory in order to grow and adapt. The purpose of theory is to organize information and accumulate experience while improving the organizing concepts. As part of our art is dependent upon abstract theory, a key role is for the university to provide relevant theory. This is efficient as we probably learn abstract things easier when younger than older. That may be a function of the point Dr. Smith made: As we get older we get busier and busier, and don't have time to sit down and go through painfully abstract notions. However, pure theory is not too useful and we need a mix of reality to demonstrate the validity of the theory. The store I work at is worried about building on undergraduate theory—adding a few concepts and exposing the student to a range of industrial experiences. We're not trying to teach everything. We're trying to teach some fundamentals well and hopefully stimulate the student to continue his education by filling in the gaps.

Now Kapur asked me to talk about what we're doing because I think we're making a mild start at the problem of joint industry/university cooperation. I'd like to give you about three examples. I point out at the outset that there are gentlemen in this room with whom we're working and that have been

strong supporters, including one we even tried to talk to come visit with us for a while.

There is a program that we mentioned called the two year program in Management Information Systems. This, believe it or not, is a joint venture between the Division of Engineering and the Business School, and is the first school cooperative venture at the Master's level in the history of the department. That in itself should identify something to you. We are appealing to people with Bachelor of Science degrees in mathematics, computer science, and physics that have a pretty decent background in analytical and abstract thinking, with two to three years of experience, hopefully in programming or systems analysis. Actually, Tom Cheatam is organizing this program. Neil Churchill and I are the business school side of it, and we have eight students this year. We wanted to start out very small. Next year we're going to have 12, and hopefully the year after, we'll go up to our eventual enrollment of 20.

Cooperative

The most interesting part of the program, at least from my point of view and yours I think, is that we have two industry cooperative activities. The first is a seminar series, which invites individuals who are managing computer systems or implementing large systems to come and spend an afternoon with the students on an *ad hoc* basis and discuss with them their problems and describe their systems. This turned out to be sort of a "show and tell" type problem. They come and describe what they're doing and then get into an informal give and take with the students. Then they normally have dinner and sort of raise the unsolved problems which were posed.

Second is the summer intern program. We have every student signed up to work in an on-going system development effort for the summer and we're trying to make it a useful learning experience by getting them to develop, if you will, a learning agenda before the summer.

We're going to discuss their progress and work experience with them during the period as well as require a follow-up report.

A second effort of which Mike Samek and Paul Strassmann have been very supportive has been a computer-based information system M.B.A. course, which essentially is a field project course. We recruited about 12 companies, I think maybe more than that, to sign up for a three-year program. This is three years at minimum, if you will, to absorb a team of M.B.A.s each spring in the analysis of some system problem in this organization. At present, I think we have 12 such teams. Last year we had ten, so the program is growing.

The students spend two to six days on site and then are expected to write a professional report, give an interim presentation on what they have found as the problem with the company management and faculty in attendance. Later they present a final review of their findings. I think it has been an extraordinarily effective learning experience for the students to go out and find that life is sort of messy and these people problems seem to dominate the theory problems. There have been a lot of neat things you'd like to do that just can't work from a faculty perspective. I think we've found it at least a tolerably decent way to keep up with the outside world and understand what our problems are. That may be also one of its prime benefits.

Case writing

A third effort, which is more traditional with the institution, is case writing. At the present time, we have four major programs. They are the executive programs, which are a series of 13 week short courses; the Master's program, which is a two year program; the doctoral program; and then, finally, international teachers. We have about 60 class days, on computer systems and management information systems, that are in some sense required for one or two of those programs. Those are primarily case-oriented class days.

To keep those fresh, we have to write on an average of 20 to 30 cases a year, because the technology changes so fast. Now Ted has been most gracious, as has Paul, in welcoming us, because to write a business case on an information systems topic requires an enormous amount of support from the company. To really understand what is going on takes about a week of somebody's time and then careful review of written statements to be sure they are accurate and define the issues. And finally, there needs to be a willingness, perhaps, to air a little dirty linen and not get upset about it because it's the issues of perhaps unforeseen events that are the important grist for our mill.

This effort on our part requires about the equivalent of two

full-time faculty individuals each year on nothing but case writing or administering these field efforts. For this we have so far been fortunate in getting industry support both for travel funds and for the necessary research assistant support.

Opportunity

I cite these as examples because I think there is a lot of opportunity, but it takes time on both the academicians' side and the industry side.

Our feeling is that the best programs have been those that extended over a longer time horizon than two to three years; because, in fact, universities and business aren't institutions, they're made up of individuals who are willing to do a few voluntary activities they understand. Understanding a field program takes time of both parties and it is important to use this understanding for more than a one shot activity.

In the computer industry, or computer business, I guess, I have to identify that because the skilled individuals who are successful and competent are not large in number, you keep running into the same few. I suspect that we can list 500 people in this industry that have had good experience, are competent in understanding the issues and perhaps capable of executing long complex experimentation. You look at the size of the industry and that is astoundingly small.

It is that resource that I think we have to share. And I think interaction by people for a purpose that will accumulate some experience is what we're going to have to develop and the unfortunate problem is that a lot of that experience ages enough in four to five years to be out of date.

Question:

What about the undergraduate level? Both of you have talked about the Master's program and I understand your requirements. Looking at it from the industry side we can only take so many Master's folks and we only have so many career paths to satisfy these folks. We also need some pretty solid citizens that have the basics of business, the basics of computer science. And from my personal viewpoint, I think you can give us some pretty solid undergraduates to handle that. Where are you on undergraduate programs?

Answer [Smith]:

From my point of view in the College of Business Administration at Texas A&M, we currently have four undergraduate degrees in business. In order to have a computer science degree we would have to get internal approval among our colleagues through the Dean, from the Academic Council, and from the Coordinating Board.

I doubt this will ever happen. Now we are able to have perhaps four to five courses that are electives, which can be fed into the traditional degrees but I don't see on the near horizon an undergraduate B.A. degree in business with a major or a field in computer science.

Comment [McKenney]:

We don't train undergraduates. First of all the Business School doesn't have any and the university in general "educates undergraduates" through a liberal board perspective, most of whom go on to graduate school. I think at last count 92 percent of our undergraduates go on to graduate school. I suggest to you that that may be an opportunity for business to rethink, in fact the culture in general to rethink, as I look at the evolution of the entrance level potential. With 37 percent of all undergraduates going on to some form of graduate education right now, it may well be that the entrance jobs, particularly in systems analysis and the management of systems, are going to require a Master.

You can only cram so much into someone's head for four years. And I guess, as a person that has to cope with them all the time, it seems to me that the maturity of the undergraduate is not all that great and it may take them to the time they're 23 or 25 to grapple with some mucky issues. I guess that my response is that we're probably not going to do anything as far as the undergraduate degree. We're putting most of our discretionary resources in the doctoral program to train teachers of undergraduate programs so that they have that mixture of computer science and management.

Question:

Jim, you suggested maybe a kind of a paradox in which it requires the Master's level to play the game but at the same time we all know that the intention of going to college, number of years spent and so on are declining. And you spoke, as I guess everybody did, about stuff becoming out of date. Does that imply moving toward a new pattern, a kind of formal academic retreading for two or three years at a time maybe throughout a long period?

Answer [McKenney]:

My own personal opinion is yes. I think we're seeing that the steel industry does it now. I feel the educational system in America and western culture was developed in an agrarian society which wanted to train its young because they were going to die at 50 years of age and they needed to be home on the farm in the summer to manage. You have a basic constraint that we live in and now we're in an environment where most of us have a life expectancy of 70 years, schools have air-conditioned buildings and we may not have students very long and the needs are very complex.

I've participated in two ivory tower think sessions where we looked at an educational system that in fact would be a recurring event every six years. The problem comes from the fact, of course, that the 27 to 35 year old age level in our culture is a time of most intense economic need and family formation and a whole number of other social actions which may preclude that in part. Though the recent statistics are that the marriage age is going up, it is a fact that the number of individuals applying for colleges as a percentage of the age classes is still increasing.

Question:

I have a couple of comments on the same subject: First, are you people going to teach these Master's level people that all systems analysts aren't vice presidents? I think this is the concern that we have. These people come out and they think that they are going to have my job in a day or two and I don't have any place to go.

Secondly, I'm concerned when we talk about computer science people. Very few businesses that I know of, mine in particular, are in the data processing business. We are in the insurance business and have to do data processing type work for an insurance company. I think this is an important aspect of the whole thing. Also a comment would be that continuing education in our company is forever and the universities have done a pretty shabby job of providing service. Now we're getting it started slowly in our area, but this is the thing which interests me more than getting more Master's level people into my area of operations.

Answer [McKenney]:

In the first issue I'd have to respond that I share your opinion, that I wish the aspirations of my students would be more realistic; but on the other hand there has to a certain optimism for someone who's 24 years old and spent $7,000 on something. I'd have to say that as far as our program goes, we spend more time worrying about defining business needs than computer problems. And in fact, as both Paul and Mike know, we rub their noses in business situations and wonder how we can meet the needs of the managers.

I think we have had a long tradition of strong relationships with industry and we have been rewarded for that. It's not easy to start. I think Smith's point, for my colleagues across the river, it's not clear that we still aren't a "house of prostitution" and we like our side of the river and they like theirs. Other universities aren't as fortunate to have had a school of business that really said "To hell with it," and went their own way. I think that having taught at Stanford, UCLA and Purdue, where you're in an overall university community very dominated by academic tradition, helped. At times it's hard to respond to industry needs and I would submit that most universities, in particular the midwest, are dominated by trustees who could influence some sort of industry cooperation.

Comment:

This seems somewhat contradictory that you are concentrating on graduate degrees, but the prerequisite for your graduate program is the individual who has been out working three years or your undergraduate program does not train him to do anything useful during those three years. I think that from science we may have gotten some bad habits. We keep a man in school from kindergarten to Ph.D. and in some fields this is logical because as he progresses, he must use what he learned last year in this year's problem. But in this field there are so many pieces to it that I think that a person can accumulate a lot of course credits and never know what the devil the whole thing is supposed to look like when he integrates it. So the idea of going out and working for awhile and coming back has tremendous benefit.

In government, we have addressed the problem of training out of necessity. We underwrite the possibility of going back to school on both a part-time and full-time basis with a very careful selection process. It's a very selfish endeavor of the organization. But, at the bottom, we need people who can come in, who are useful and who haven't got such a huge investment that it's only logical for them to demand high pay and high positions. It's very inconsistent with the real world.

Comment [McKenney]:

Don't you think the economic market can dictate that? On the subsidy problem, I'm the head of a group that is looking at Russian management education. They have this on-going process and they do subsidize. Every student gets 40 rubles a month and if you do well in class you can get up to 100 rubles a month. Their "computer science" degree is in management. Their management is essentially a combination of industrial engineering and cost accounting. It is a 4 1/2 year program, with the last six months in residency in industry, working on a system development or project, and they go out for three years. They expect then to come back the fourth or fifth year for another one or two year period. I don't know that we'll ever get to that planned environment where we could do this, but I do think the market factors are right.

The job markets are beginning to pressure San Jose State and a lot of other institutions to understand that if they are going to attract students, they are going to have to provide the student with a means to a livelihood. Now whether you philosophically enjoy that, that seems to be the way it is happening right now.

Comment:

In terms of the basic students, it would be difficult to take them out of their teaching environment or to try to spend a few hours of that 60 hours each semester on painting a picture of the environment for them. We must be careful that they don't come out thinking that the world revolves around programmers or systems analysts. They think it very odd if somebody says go talk to the comptroller about whether we can afford that. Or the programmer may have studied your business case, but when he gets in the firm if he's in the data processing department, the case requirements will be dictated to him by the customer. He can get in big trouble very fast by fooling around with the customer's rules of business transactions.

Question:

I must say that I agree with the gentlemen about the undergraduate work. If we're going to address the problem that Dr. Smith mentioned at the beginning of the session as to the proliferation of computers and the requirement of these in our business, I certainly think we ought to see that we are going to need trained people. I don't see how we can possibly have enough Master's graduate people to fill the need in the time frame that we're talking about. By necessity, we, in the business world, are going out and pulling in people from engineering schools, which are the only schools that offer the undergraduate computer science degrees. And I wonder, why can't we change it, Dr. Smith?

Answer [Smith]:

You have a lot of clout and you don't know it. And if you people in the room would use your influence collectively, you could make a lot of changes which are needed. We need your ideas and your influence. You should provide academia with input as to your needs, then help convince the people in authority that these needs are valid.

Question:

Who do we convince?

Answer [Smith]:

Well, to start out with the deans of the various colleges of business. Now it will have to go further than that. You know eventually the coordinating board or some similar agency has to approve in the case of new degree programs. But I think this is getting right down to the problem.

Comment:

That is what we are here for.

Comment [Glaser]:

Would Paul Strassmann (Xerox) and Mike Samek (Celanese) care to comment on the value of Harvard's field project course on information system for M.B.A. that Jim McKenney described?

Comment:

It's valuable to the students but certainly it enables the people working with students to have an understanding that coming from Harvard are some very well educated people who will be able to quickly float to the top of the established organization. The interaction with the Harvard students has spurred our internal people to go out and get their own dose of education. As a matter of fact, there is a correlation between Harvard M.B.A. visits and the kind of applications I get for local M.B.A. subsidies.

Comment:

Well our experience has been slightly different. We generally support education anyway, within the company. Lots of people go out and get support for graduate work. The experience has been interesting to us. We tried to make this as useful to the students as possible.

I think that, in this particular program, the time demands on the students are so unbelievable that they're barely able to squeeze in what they really would like to do.

Section Two: What Industry Thinks It Needs

The second panel session was on "What Industry Thinks It Needs," and was led by Mayford L. Roark, executive director, Systems Office, Ford Motor Co., Dearborn, MI. Other panel participants included Harlan Mills, IBM Federal Systems Division, Gaithersburg, MD, and Robert Ott, national DP manager for Sears Roebuck Company, Chicago, Ill.

Questions which were asked after the presentations are included at the end of this section.

Mayford Roark

This workshop has provided a welcome opportunity for serious thought about what we in industry really do expect of the universities in the field of systems education. Too many of us, I am sure, see the university as a sort of screening and sorting institution--a place where promising young people can be brought for a probationary period of testing and grading. From that kind of limited perspective, it wouldn't really matter what was taught as long as the end result was a careful identification and grading of the best available recruits. We at Ford, as with many other companies, want the best talent we can get; so we do give careful attention to grades and evidence of leadership. We are also interested in the substance of an applicant's background.

Systems education has suffered because we haven't quite come to the stage of defining the systems function as a profession--or even establishing whether it is a profession. Meetings like this ought to bring us closer to resolving these uncertainties.

Even to speak about the systems needs of large manufacturing organizations is a considerable task, so in preparing for this meeting I sought the views of my colleagues at Ford. The summary of their views that I shall present was derived through a questionnaire that was completed by 21 systems managers who expect to be hiring people this year. Each of these managers is responsible for a systems activity in some division or staff component of Ford.

Kinds of jobs

One of the problems that we face in this kind of discussion is a lack of uniformity in vocabulary. Let me begin, then, by identifying the different kinds of professional systems jobs that we see at Ford. The categories of people I want to talk about today are indicated on Table 1.

The 1,000-plus people covered in this tabulation comprise most, although not all, of our "professional people" in North American systems activities at Ford. The ones not included belong to smaller activities that do not plan any hiring this year. The listing does not include classifications such as computer operators or data transcription personnel; I am assuming these to be outside the professional mix of interest to the universities.

Within the professional groups listed, the largest categories are Programmers and Analyst-Programmers, each accounting for 29 percent of the sampled population. Another 18 percent consists of Systems Analysts. Some of our larger organizations find it desirable to maintain the Programming and Analysis functions as separate entities. Other organizations prefer the hybrid Analyst-Programmers for handling both functions.

Our count of Operations Research Analysts is surprisingly small in view of the rather large attention this function has been given academically. Even with only 3 percent of our systems people involved in the OR function, I believe this may be a higher-than-average proportion as compared with most other companies.

The Technical Specialists (5 percent of the sampled population) are principally software people, although there are also a few specialists in computer hardware and systems measurement. The remaining population of su-

pervisory personnel is not of direct interest for this discussion; we would not normally expect to recruit such people direct from the universities without a period of experience.

Hiring trends

The recent trend of our hiring experience is as indicated on Table 2.

This year, the managers in our sample indicated they plan to hire 105 people, of whom roughly two-thirds are likely to be recent graduates. In line with the trend of recent years, 82 percent of the planned hires will go to work as Programmers or Analyst-Programmers. The remaining handful of recruits will be about evenly divided among Systems Analysts, OR Analysts and Technical Specialists.

The principal point of interest in the history displayed here--other than the fact that 1975 was a bad year in our industry--is the trend *toward* a greater mix of Programmers and Analyst-Programmers, and *away* from the broader category of Systems Analysts. I interpret this to mean that, as the development and maintenance of systems becomes more complex, the need increases for people with a good grounding in the technical fundamentals of the business.

Desired education levels

One of our first questions had to do with the desired educational levels for the different categories of recruits. Table 3 shows the profile of responses.

None of our managers was interested in hiring for any of these skills below the Bachelor's level. Roughly half of the hiring is projected at the Master's level, with about a 50-50 spread in the case of Programmers and Technical Specialists between Computer Sciences and other majors. Systems Analysts and OR Analysts, for the most part, are viewed as coming from tracks other than Computer Sciences.

Table 1. Categories of Ford Systems People at Sampled Activities

Category	Headcount January 1976	Percent of Total
Programmers	305	29
Analyst-Programmers	311	29
Systems Analysts	190	18
OR Analysts	30	3
Technical Specialists	60	5
Supervisory	171	16
Total	1,067	100

We can't, of course, recruit people solely on the basis of qualifications for their entry jobs. We need to have some ideas about their future possibilities. There are several reasons why we are particularly attracted to recruits who have long-range potential for general management.

1. Our influence as a systems community within our company is enhanced as our alumni go on to greater things in other areas. An encouraging number of our systems people at Ford have gone on to senior executive positions. We count their success as our success.
2. In the sociology of large organizations, the rewards go to managers more than to technicians. The talented people we want to hire, therefore, must perceive an opportunity

Table 2. Ford Systems Hiring Experience, 1971-76

	1971	1972	1973	1974	1975	1976 Est.
Total Hires:						
Worldwide	202	232	312	153	34	160
Sampled Activities						105
Professional Mix (in percentages):						
Programmers, Analyst-Programmers	53	60	62	67	94	82*
Systems Analysts	27	28	29	25	0	5*
OR Analysts	5	6	3	4	0	7*
Technical Specialists	9	4	2	3	6	6*
Supervisory	6	2	4	1	0	0*
Total	100	100	100	100	100	100
Trainees as Percent of Hires	37	42	43	44	33	63

*1976 detail is for sampled activities only

Table 3. Desired Educationals for Ford Systems Recruits

Category	Percentage of 1976 Recruits: Bachelor	Masters, Computer Sciences	Masters, Other	Total
Programmers	66	17	17	100
Analyst-Programmers	44	28	28	100
Systems Analysts	36	0	64	100
OR Analysts	0	33	67	100
Technical Specialists	0	50	50	100
Total	49	22	29	100

for advancement or they will find their careers confining and frustrating.

The most important sign of management potential is communication skill--both oral and written. In the questionnaires, I did not ask about these skills, but several managers emphasized them anyhow as being the most important attributes they will look for in hiring people. Without doubt, these communication skills are the major determinants of who, among our recruits, will adapt to higher levels of responsibility either in or out of the systems function.

Secondary backgrounds

We asked whether there were any preferences about the kinds of backgrounds desired other than formal systems training. The result was a rather emphatic preference that the candidates should have such secondary backgrounds; only 20 percent indicated no preference. See Table 4.

The kinds of secondary backgrounds desired reflected mostly the organizational needs of each manager. Managers in the marketing organizations and staffs tended to prefer Business Administration backgrounds. Some of the manufacturing divisions preferred candidates with backgrounds either in Engineering or one of the sciences. Several managers commented that past experience with Computer Sciences people had been unfavorable because the individuals had lacked any understanding about organizational disciplines such as Management and Accounting. About a third of our computer application programs deal in one way or another with financial problems; a Programmer without some formal training in Accounting or Finance would undoubtedly be handicapped in facing such an area. A Programmer in some manufacturing divisions might have difficulty facing his Manufacturing Engineer customers unless he had some engineering background. See Table 5.

Table 4. Desired Educational Backgrounds

Category	Percentage of 1976 Projected Recruits: Business Admin.	Engineering	Sciences	Liberal Arts	No Preference	Total
Programmers	40	13	21	5	21	100
Analyst-Programmers	38	10	3		49	100
Systems Analysts	33	42	17		8	100
OR Analysts	100					100
Technical Specialists	17	49	17		17	100
Total%	41	17	13	3	26	100

Types of courses

We followed up on this point by asking specifically what kinds of Business or Engineering courses were viewed as "essential" or "desirable" for the different categories of recruits. For Programmers, only a few of the managers were willing to single out any particular courses as "essential," but a strong majority listed courses like Accounting, Statistics, and Management as being either "essential" or "desirable." The more technically-oriented managers expressed a similar preference for Engineering backgrounds, with Industrial Engineering being somewhat more in demand than Electrical or Mechanical. For OR Analysts, there was an emphatic demand for basic business courses.

We asked about specific courses that might be desirable in the Systems area. The results are summarized in Table 6.

Several items stand out in Table 6.

1. Even for Programmers, COBOL was not viewed as being a universal requirement even though it is our standard

language for business application systems. One of the managers explained his view this way:

"Proficiency in any particular language is relatively unimportant, but proficiency in some language is important because a second language is much easier to master than the first. I would consider a student who is proficient in APL or PL-1 to be a solid candidate even though we do not use either language."

2. Aside from programming languages, the strongest course preferences were for what we called Business-Systems Analysis or MIS, Structured Programming, Data-Base Management and File Structures. These were viewed as being fundamental working tools of the Programmer or Systems Analyst.
3. The profile for Systems Analysts was not much different from that indicated for Programmers, although the preference proportions were a shade less emphatic. In general, our managers view Systems Analysts as needing the same basic working tools as those expected of Programmers.

We asked a similar question with respect to some courses commonly found in OR curricula, with the results shown in Table 7. These courses were obvious staples for the OR diet. Some were also seen as desirable background for Programmers or Analysts.

Assessing the results

I have tried to relate these responses (which seem to me an accurate assessment of our needs) to what I know of the offerings at universities where we recruit. Here are some of the questions that come to mind:

1. Why shouldn't some basic courses in Business Administration--like Accounting, Statistics and Management--be an essential part of a Computer Sciences background? (They do seem to turn up in the MIS curricula, but they almost never appear in Computer Sciences.)
2. In a similar vein, shouldn't there be a course, or part of a course, on "Designing Systems that can be audited?" Computers do most of the check-writing in the United States today, as well as most

Table 5. Preferences for Business and Engineering Courses in Percentages

	Programmers and Analysts-Programmers		OR Analysts	
	Essential	Essential or Desirable	Essential	Essential or Desirable
Business Courses				
Accounting	19	86	50	100
Statistics	14	90	100	100
Management	5	62	50	100
Engineering Courses				
Electrical	5	24	0	0
Mechanical	0	19	0	0
Industrial	0	33	0	0
Memo: Respondents		21		2

Table 6. Preferences for Systems Courses

	Programmers and Analysts-Programmers		Systems Analysts	
	Essential	Essential or Desirable	Essential	Essential or Desirable
Programming Languages:				
COBOL	52	95	0	75
FORTRAN	10	57	13	38
Business-Systems Analysis or MIS	19	81	38	75
File Structures	10	67	25	75
Data-Base Management	5	76	0	75
Structured Programming	0	86	13	88
Data Communication	5	57	0	50
Major Applications (Finance, Supply, etc.)	0	62	13	50
CAD/CAM Technology	5	14	0	25
Minicomputers	5	43	0	50
Memo: Respondents		21		8

Table 7. Preferences for or Courses

	Percentage of Respondents		
	OR Analysts	Programmers and Analysts-Programmers	Systems Analysts
Simulation	100	43	50
Inventory Management	100	43	38
Production Management	100	43	13
Optimization, Math. Programming 100	38		25
Forecasting	50	38	25
Memo: Respondents	2	21	8

other financial-control functions. Understandably, computer systems have become a prime target of the Audit function.

3. Some of the courses we do see listed are not really clear as to their objectives. What benefits should I, as a prospective employer, or the student as a prospective recruit, expect from courses like "Adaptive Systems," "Theory of Automata," "Theory of Natural Language Structure?" I don't mean to criticize these as course offerings--I'm just asking where students can find time to take these courses in a Masters program that is too crowded to allow for Accounting and Statistics or for working tools like Data-Base Management.

4. Are the universities being unduly inhibited by concerns that their offerings in this field will not be academically respectable? Do courses have to be formulated at a high level of abstraction in order to be perceived as scholarly? If so, why should this be more true for Computer Sciences than for some other fields that are bursting with courses related to specific vocational functions?

5. Can't more be done to teach students how to document a system, both for users and for technical operations, in clear, understandable English and appropriate top-down charting? Can't they be taught how to write reports, prepare presentations, and develop systems proposals? Can't they be better prepared in the communication skills on which their future advancement will so heavily depend?

6. Is there enough emphasis on the "Economics of Computer Systems?" What are the trade-offs that justify the development of a System? When is it better not to computerize a function? How can a problem be evaluated in terms of optional approaches or strategies?

One other thought, from the point of view of the student: If a young person invests two years and upwards of $10,000 in a Computer Sciences Master's degree, how should he feel if he discovers some years later that his advancement through a systems career in industry is limited because he was never exposed to the basic working tools of business? One of our brightest young people--a Master's graduate of a Computer Sciences program at one of the prestige schools--sent me the following comments after reading this paper.

"I can identify each of your points with an inadequacy in my own education. My own program included none of the courses in Business, Accounting and Economics that are so necessary to the solution of a systems problem in its natural context. The lack of these basics has always placed me at a disadvantage to others who had a more rounded education."

I realize that these concerns may reflect a limited view of industry needs. Even so, the interest that is evident in this problem through the participation of so distinguished a gathering from industry, government and the universities is a heartening indication that our concerns are widely shared and appreciated by others. We are grateful to the DPMA Educational Foundation for making such a meeting possible.

Robert Ott

I have been asked to comment on the needs of the retail industry relative to new data processing employees. Let me begin by saying that in my 12 years experience in data processing, I have never seen the demand for data processing talent any stronger than it is today. Technology is changing faster than ever, and new data processing professionals are vitally needed to handle the rapidly growing demands of the total business community.

I don't believe that the needs of Sears and that of the rest of the retailing community are really very different. Like other retailers, we are looking for the best technical talent with the best social skills all wrapped up in the same individual. If we hire only technicians, we will suffer within our systems and management ranks in the future years. If we hire only socially adaptive people, we will not have the technical expertise to carry on the very complex data processing function. And so we search for the individual capable of handling both sides of the spectrum.

Of course, that is being a bit idealistic. We may search for this ideal candidate, but we usually must be satisfied with something less than perfect. Thus, quite automatically, we arrive at the rationale for our training program. We take the products of our recruiting efforts and attempt to build social skills into our technicians and technical skills into our socially adaptive people. The technical skill training is emphasized during one's early career years as a programmer, and the social skills are stressed as one moves into areas of project responsibility and client interaction.

This has been generally true throughout the history of our data processing department. I believe it would prove helpful to trace this development and see how our program has had to adjust to properly prepare college graduates for entry level data processing assignments at Sears.

Growth in beginning

In the early days of our department, we experienced tremendous growth in data processing requirements. The very large operations of our company were being computerized, including credit, order processing, accounting and merchandising systems. During that time, the campuses were offering one or two courses in data processing which served as an introduction, at best, to our data processing endeavor. System 360 was being introduced. And we, like many others, were moving into a new era of data processing systems.

The advantages gained from the new computer generation did not come without its share of problems. It was necessary to retrain our staff in the new 360 assembler language. In addition, our staff had to be exposed to a new, more complex operating system. This was done in the quickest, most effective way available at the time. Programmed instruction manuals were used to teach the language and our own personnel from the software group were used to teach full day classes in the basic uses of Job Control Language and the operating system. At best, the initial products of this training effort had a general understanding of the processes and procedures of S/360. Understandably, however, a firm grasp of the system and how best to use it were not being presented.

Because of such gaps in the technical knowledge of our staff, it was also difficult to determine whether a slow and inaccurate programmer was inept, or whether that person simply had not yet developed good programming habits. Resident vendor experts were available for consultation if someone had difficulty with a program, but obviously programmer productivity was not high during this period of early development.

In-House Training

In this environment, and as the size of our staff passed the 200 mark, we realized that something had to be done. In the Spring of 1968, we decided that the only answer was to begin the development of in-house training for new data processing employees. One person was assigned to work full-time with the new trainees who were continually being added to the staff. Our initial objectives were relatively simple, but obviously crucial:

- First, to teach the programming language using the PI books.
- Second, to have each trainee complete one or two simple programs in order to learn basic programming techniques.
- Third, to provide for attendance at available IBM classes on the operating system and learn to use the material presented.

The programs of each trainee were reviewed by the training coordinator for both accuracy and good coding practices. Only then were they moved into an application group where they would begin productive programming. This insured at least a basic degree of proficiency.

One immediate benefit of this approach was that we discovered rather quickly which trainees were not capable of completing the required work before they were committed to an actual project team. Over the years, this has helped in upgrading and maintaining the quality of our professional staff.

Nevertheless, within the first six months it became evident that, even though our trainees were attending free vendor-supplied classes, they were learning only the very basic concepts of how to flowchart, assemble, code and test their programs. Indeed, our training coordinator had to conduct individual sessions with each trainee to assure their understanding of these principles. It was not very long until we determined that it would be necessary to reach these subjects internally in order to achieve a greater level of competency for our students.

Flow charting

The first flowcharting and JCL classes were taught in late 1968. This effort was so successful that it was expanded in early 1969 to classes which covered not only JCL, but O/S topics which a beginning programmer needed to know.

Finally, in 1969 a complete in-house class was developed that provided a basic but comprehensive introduction to data processing. Materials were developed to supplement the programming language PI course, and our practice programs were refined to cover more programming techniques.

Today the basics program continues to produce qualified graduates who are capable of working in the highly technical assembler language environment in our department. Of course, there have been some changes in format. We no longer use the PI books to teach the language but, rather, a combined format of lecture and workbook assignments. In addition, we initially introduce our students to machine language and then work up to the concept of assembler language.

In late 1969 and early 1970, we began to see a need for internal advanced technical training. The combined effect of our proven entry level training program and better computer courses on the campus were resulting in a programming staff which could handle the more advanced concepts of operating system and programming techniques. We began with a course in modular programming and followed with courses in macro-coding techniques, intermediate O/S topics and a full 7-day O/S workshop.

Today, our advanced programming staff teaches courses in modular programming techniques, access methods, improved programming techniques and CICS.

Technical expertise

As I mentioned earlier, technical expertise should be supplemented by well developed social skills. In order to provide training to develop our non-programming skills, we currently offer classes in systems analysis, systems design and effective interviewing in the systems environment. We are currently developing courses in structured design and project management. In addition, our corporate personnel department conducts classes for us in effective communication,

managing change, and management in action. Supplementary to this list of lecture and workshop courses, we provide a videotape training program featuring courses in PERT/CPM, improved reading, transactional analysis, business writing, cost benefit analysis and utilities.

Viewing the campus

Enough said about the training available to our people. As we view the campus today, what kind of people are we looking for? Frankly, we are seeking about the same candidates today as in the past. We recruit candidates from all majors. We recognize that a few courses in data processing might be helpful to indicate a definite interest and basic aptitude. But beyond that, we have no firm or definite computer science requirements.

It is a prime requirement to find people on the campus who have the maturity and intelligence to function within our highly competitive and rather demanding data processing activity. We do not want to limit our search to computer science majors alone, since we always want to consider the best individual. Thus, with a comprehensive training program backing us up, we can be free to seek the best qualified person available. This is not to say that we bypass the computer science major. Instead, they are considered with all others and evaluated as to their degree of social maturity and their degree of technical knowledge.

This approach results in a wide range of educational backgrounds within our training classes. In fact, we have not found significant variation in final job performance between computer science majors and personnel we have hired from other disciplines. In many cases, non-computer science majors have done somewhat better, both in the training program, and beyond. This multiple major approach to recruiting also has been a useful policy in our current program of affirmative action requirements. Since there are insufficient minority candidates in computer science curriculums, we have the ability to seek minority candidates from all disciplines within the university.

By now you may have asked yourself, "If Sears did not have an entry level training program, what would a student be required to learn on the campus?" To answer this question, it would be necessary for the student to have a working knowledge of IBM assembler language, a thorough knowledge of basic JCL, an introductory understanding of the operating system and a working knowledge of basic IBM/TSO.

Beyond entry level

What about beyond the entry level subjects? Would it be advantageous to train the campus graduate in subjects such as data base design, advanced access methods, minicomputers, advanced macros, design of compilers, telecommunication or other advanced technical subjects?

Frankly, we do not feel that this kind of training is that beneficial to our entry level candidates. Our rationale is simple. Before we move any of our employees into more technical activities, we want to be sure to expose them to some type of applications programming. The one thing that a new programmer knows nothing about is the workings and procedures of Sears. The application team is the place to learn about the company and the basic disciplines of the data processing profession.

Subsequent to initial exposure in one of the applications areas, it is possible to move the programmers into more technical areas. But, by this time, the previously learned campus knowledge may be obsolete. Thus, we see the need to teach such advanced concepts internally, so the employee is kept up to date with current techniques and technology.

Of equal and sometimes more value, to my way of thinking, would be a strong business background, with emphasis in some technical aspect of business such as statistics or accounting. This might better prepare the student for an understanding of the business environment, along with regiment necessary to function at a rather tedious detailed technical desk job.

Beyond this, a thorough experience with group interaction, the systems development life cycle, systems design and life under a strict data processing production schedule would be welcome additions to a candidate's background. And very important, it would help if college graduates could write English clearly. There is little we do in the business world that does not have to be communicated to someone, and much of it requires written form and documentation. Yet, our college graduates and business people in general are poor writers. If it would accomplish the purpose and I had the option, I would require four years of English in all colleges and universities.

I don't want my remarks to sound too critical, as I do believe our educational institutions are doing a pretty good job of preparing candidates for business. But I must say, they are not doing a complete job. Consequently, we have found it necessary to supplement the educational process with a rather substantial internal training program for data processing people. It may be necessary and best for us to continue our basic and advanced training program in data processing for some time to come, and just look to the universities for well educated, socially adaptive graduates who have good aptitude and interest in the business environment.

Perhaps we are recognizing that it is a joint venture between the universities and business. The educational institutions should teach fundamentals, and we in business must bear some responsibility for additional and ongoing education in technical areas.

Harlan Mills

What I'd like to do, just from hearing what's been said before, is to talk a little bit in behalf of mathematics. Mathematics, to me, is a synonym for thinking straight. I don't mean higher mathematics, I don't mean powerful mathematics in the sense of differential equations, or topology.

About myself

It may be helpful to be just a little bit autobiographical. I started out as an English major. At the age of 27, after World War II, I came back and started out as a freshman in mathematics. It changed my life, and I got all my degrees in mathematics. I got my PhD. at Iowa State, which is an applied mathematics school; and teach at Princeton, which is a pure mathematics school. So I had three educations: One in business (during my twenties mostly, before the War); second in applied mathematics; and third in a pure mathematics.

But let me come back to something a little bit more manageable first. Talking about the needs for some new kinds of mathematics, it's sort of like the new math, although there's lots of others in computer science in areas like formal grammers, automata theory and so on. I'd like to add the case and give you a couple of anecdotes for those needs.

I'm not talking about a course in automata theory. I'm talking about two weeks to find out how finite state machines work, another two weeks to find out what formal grammars are and how they relate to finite state machines, and other short shots in a larger program of applied computer science that a business school graduate could make use of.

Power of math

I'd like to tell a couple of anecdotes which I think represent the power of mathematics in practical computer affairs, and to illustrate this power by the kind of things we should teach.

Analysis

IBM developed a system several years ago for *The New York Times* to automate *The New York Times* morgue, which is accessed through *The New York Times Thesaurus*. The description of *The New York Times Thesaurus* was over a dozen pages of two-column fine print, written just like a lawyer would write a contract; this is the way you get in and out of the thesaurus. It was important to the project, because the thesaurus occupied seven 2314 disc drives several hundred million bytes of material which *The New York Times* reporters were to browse in.

And so we are pondering at the beginning of that project—how can we program down the left-hand column of page 7 without the right-hand column of page 9 biting us?

Well, a couple of days work on that thesaurus, all that nice, beautiful, legal-like English, produced a formal grammer (in Backus Normal Form). In 15 lines we described the entire syntax of this thesaurus. Once we had that, we now knew that the material on page 9 isn't going to bite when we program on page 7, because we've got all the logic in front of us in 15 lines of grammar. I hesitate to use the word mathematics, I use it to talk about thinking straight.

To be able to reduce such problems to that extent by using mathematical notation is what we need to teach. In two weeks, we can teach students about building formal grammars. For example, the sequences of all possible bids in bridge can be described in formal grammar. Most students in the computer science curriculum think that formal grammars are good for describing programming languages. Well, they're all right for that, but that's just a drop in the bucket compared to what formal grammars are good for. They're good for describing files, inputs, outputs, and so on.

Design

The second example deals with design of *The Times* on-line conversational system.

The Times system is designed as a collection of nested finite state machines, about 50. For example, 'SIGN ON' is a finite state machine. 'BROWSE' is a finite state machine.

Many of these finite state machines are broken into submachines, so that, for example, 'SIGN ON' itself will consist of a set of maybe a dozen sub-finite state machines.

The ability to design with mathematical precision allows you to then later code with a great deal more confidence and precision, too.

Now, these are things that are easy to teach and easy to learn. And once our students get them, they become much more powerful. Not a lot of mathematics is needed, not a course in automata theory, or formal grammars; just ten pages can do wonders for a student. I'm sorry that somebody hasn't collected all those groups of ten pages from here and there and put them together—I'm sure that someone will eventually—but that kind of thing seems important for the business student.

Again, I'd like to emphasize that we don't need to talk about computer science. We need to talk about good management practice; and good management practice involves divide and conquer. It involves organizing complexities. How does one decentralize a business or decentralize an operation? Just divide and conquer, just like you'd design a large system.

I say it because I have respect for the complexity which management undertakes. The reason I think that management understands ideas like structured programming faster than programmers understand it is because management typically understands the need for dealing with complexity more than programmers do. Programmers that come in at an entry level and learn to program a few modules (compute square roots and things like that) soon get the idea that they can do everything. You put 40 of those people together on a project, they discover that life gets very complicated. But they don't discover this during the project, they always discover it at the end when they try to make everything work together. So that the issue of dealing with complexity is something which is natural to management and, in fact, natural to mathematics.

There have been several references (to change the subject) to English. I was in Japan last fall, and one of the Japanese systems people got up and said, the important thing for our programmers to know is not PL/1, nor JCL— it's Japanese. And that's the case here.

I'm prejudiced that way. As I said, I started out in life as an English major and I have a common practice of teaching in the university; for example, I teach at John's Hopkins and Maryland both. We talk about writing good English the very first day in every course. And then we talk about mathematics as a specialization of English. In fact, most students in mathematics, even the graduate students, don't really know what the idea of a mathematics proof is—they think it's something and beyond human error rather than about a human conversation, which is supposed to be a repeatable experiment between two people.

One of the reasons that design languages are so important in the development of programming systems, is that if you hand people a programming language and you hand them just plain English, I mean sloppy English, and you say to them, "Think before you code;" they will think for a week or two about their problem, and then the urge to write something precise overwhelms them. They think, "I've got the idea now."

They'll write down a few PL/1 declarations. All of a sudden they'll be coding before they realize it, because they've had the urge to write something precise—not because they wanted to code. And in that urge to write something precise, we need to be able to teach the use of mathematics as a precise form of English.

Well, I use the word "mathematics" naturally and I've heard myself say it more than I intended to say it, because I'm not trying to promote mathematics *per se*—I'm trying to promote straight thinking, and trying to recognize that the kind of problems that we're trying to solve in society, particularly managing big systems, evolving big systems, maintaining big systems and complexes of data processing systems, are very demanding problems. They impose demands on us that we've never had before and we need all the thinking straight we can get.

The following is an excerpt of a document Dr. Mills distributed to the attendees at the workshop.

1.2 How To Develop Large Reliable Software Systems

The Problem of Scaling Up

Logically, there seems little difference between a small program and a large one. They both use the same instruction sets, the same compilers. So with 10 times the effort, why can't a program of 10 times the size be built? F. P. Brooks discusses this problem in the *Mythical Man-Month—Essays on Software Engineering* (Addison-Wesley, 1975) in various aspects. Part of the problem of scale up is that the problem usually isn't just a simple one of scale up, after all.

Brooks discusses the separate ideas of a program, a programming system and a programming product. A *program* satisfies some specifications to produce a transformation of data. A *programming system* is a collection of programs and a collection of data sets which, in concert, satisfy some user or organizational needs on a continuing basis such as maintaining a set of accounts payable/receivable that are effected by daily transactions and admit periodic and demand report requirements or maintaining an inventory of all artifical satellites on the basis of information forwarded by a set of worldwide tracking centers. Brooks estimates the difficulty of producing a programming system over a program of equal size to be 3 to 1 in effort required.

A *programming product* differs from a program in being warranted and used by a number of organizations as a standard product. Typically such a product must be parameterized or customized for different users, but developed so that maintenance and improvements can be carried out across the board. Again, Brooks estimates the difficulty of producing a programming product over a program of equal size to be 3 to 1 in effort required, independently of the programming system factor.

Thus, a *programming system product* is 9 times as difficult, size for size, by Brook's estimations. But the remarkable thing about this increase in effort is that it is just the same kind of effort—specification, design, coding, testing, documentation. These may seem like large factors, but they are borne out, over and over, in practice.

Even within a single category of program system or product, scaling up size goes up faster than linearly in effort, as a little thought substantiates. The number of possible connections among "n" items is "n(n - 1)/2" and it seems reasonable to expect program interactions to tend to such an "n2 law." So there is more logical designing and checking to do per unit of program developed. Further, as this work goes up, more people are required to do it, and to coordinate their efforts, they must communicate with one another. This means the "n2 law" again. So, as more people are added, each spends more time communicating and less time producing. Program organizations and people organizations are devised to combat these "n2 laws." They are partly successful, probably to the point of limiting these joint effects to a near "n2 law" instead of a "n4 law." The exact effects are not really known, but the fact of these effects are undeniable.

The result of the scale up is well documented in failures in large software system development—some monumental in size (tens of millions of dollars), in incredibly low productivity rates for other projects (3-5 net instructions per man day of effort), in untold human waste and frustration.

On examination of these problems of scale up, the difficulties show up at system integration time—putting it all together. There is seldom difficulty in providing a suitable design of noteworthy promise, and there is seldom difficulty in programming the pieces, the modules. The main difficulty is that the modules seldom all run together as designed. An additional difficulty (as if integration weren't enough!) is often that when the system does finally run all together as designed, it doesn't do what the users had imagined it would. So an additional problem of specifications and requirements analysis shows up that should have been handled at the outset but wasn't.

These difficulties are not due to lack of trying by intelligent people. The best people in the industry have had a nearly universal experience in such frustration and misery. In fact, it has been said that every major software system is produced by amateurs, because no one is willing to go through such an experience twice.

Top Down Development

The necessity of top down development in large software systems was borne out of bitter experience with bottom up development. In top down development, the control programs that integrate functional modules are written and tested first, and the functional modules are added progressively. In fact, the development proceeds on an incremental basis, level by level, with testing and integration accomplished during the coding process rather than afterwards.

In retrospect, it is easy to see that the advantage of top down development over bottom up development is the advantage of a closed loop feedback process over an open loop process. In a bottom up development, the modules are not tested as part of the final system until the end of the development; in top down development, they are tested in their system environment the next day. If there are coding errors of system wide (but not module wide) effect, top down development discovers them early, when freshly coded (and the original code is on hand). If there are design errors, top down development forces their discovery and corrections during coding, whereas bottom up development usually leaves them undiscovered until integration time, when both designers and coders are often departed.

Top down development is more difficult to design for than bottom up development, but the extra effort in design is made up many times in coding, integration and testing. The problem of design in top down development is not only how the final system will look, but also how the system under development will look at every stage of its construction. Building a bridge illustrates this idea. In drawing a bridge on paper, a top most girder can be drawn first, to hang in mid air until other members are drawn later to support it. But to actually construct that same bridge, a construction plan is needed, which allows girders to be placed and pinned one by one, in support of one another, until the bridge is completed.

Building a software system bottom up is like building a paper bridge, no construction plan is needed—only the final design and everyone hopes it all was together as planned. If people were infalliable—especially designers—no construction plans would be needed, but people aren't.

Building a software system top down is like building a real bridge. Finding a proper top is a significant technical task. A proper top is one that executes as a system early in the development, and which provides the basis for adding intermediate and final modules in a continuous code/integrate/test interaction process.

In top down development, design performance is crucial. It represents thinking and problem solving before coding and integration, rather than afterwards. Conversely, top down development forces design evaluation by the ongoing code and integration process. In bottom up development, poor design is often hidden until late in integration, after much functional code has been written and tested, only to be discarded.

In programming systems, top down development starts with a design of harmonious cooperation of several programs through access to several shared data sets. For example, a financial information system may include a file maintenance program, several data entry programs which produce transaction files for the file maintenance program, and several data retrieval/report programs which access the main file. Although each such program can be designed and coded top down independently, the top down system testing requires coordination between them, such as data entry programs providing input for the file maintenance program, which in turn creates files for data retrieval programs.

Design Methodology

The thinking required for top down development—"divide, connect and conquer"—is significantly more exacting than the thinking required for bottom up development—"divide and conquer." And the design methodology required reflects this additional need. For bottom up development, English plus a programming language may be sufficient. But these are not sufficient for top down development—there is simply too large a gap between English (even good English) and programming languages (even high level programming languages).

Faced with such a gap, programmers are forced into coding

too early—making detailed decisions about data strategy, about program structure too early. So a spectrum of intermediate language facilities are needed, from the flexibility of English to the precision of programming languages. We use the generic term *process design language* (PDL) for this spectrum. The demands of large scale development of reliable software systems also call for a flexible battery of auxilliary methods of logical expression—data structures, sets, formal logic, formal grammars, finite state machines, etc.—for use in describing critical systems aspects with precision and economy.

In a real sense, top down development requires the progressively detailed execution of design. High level programming languages, such as APL, SETL, SNOBOL can be regarded as executable design languages in certain situations. But any reasonable programming language, such as PL/I, can be improvised into an executable design language as well, for even broader situations. Better executable design languages may be forthcoming, and almost certainly it is a plurality of such languages that are needed, not simply one best one. Usually, the programming language of the project can serve as the executable design language, often as a final step from a more general language such as PDL.

Section Three: Opportunities for Industry/Academic Interaction

The third section is the presentation by Frederic G. Withington, senior consultant, Arthur D. Little, Inc., Cambridge, MA. titled, "Opportunities for Industry/Academic Interaction."

The questions and answers plus the discussion which followed his presentation are also included.

Frederic Withington

The word "interaction" implies individual people communicating with one another, so the opportunities for industry-academic interaction are as varied as the spectrum of individuals involved. Many forms of interaction can and should emerge; probably many will emerge that do not occur to the participants in this workshop. Certainly I will not think of them all; I am sure I do not know of all the kinds of interaction that now exist. The following list should be helpful to start the discussion process, however. It includes the most widespread methods of interaction now existing, and suggests a few more which (as far as I know) do not now exist but ought to work. It is organized under three topic headings:

1. Industry on the Campus
2. Faculty and Students in Industry
3. Desirable Changes in Behavior

1. Industry on the Campus

1.1 Borrowing Individuals

There are many examples of individual managers and specialists serving temporarily as visiting faculty, usually developing and administering one-term courses in their specialties. In a few cases, business executives do this. More often technical specialists are called upon to teach the first version of a course in a new discipline. (Data processing consultants from Arthur D. Little have taught term courses at Northeastern University, the Sloan School and the Harvard Business School.) This is healthy, but risky; the visitor is usually not a trained pedagogue and cannot conveniently be controlled. Also, he may find it difficult to maintain a pace of adequate preparation and attendance for an entire term. There may also be questions of compensation: The individual's employer may not wish to pay his full salary during an extended absence.

Guest lecturers are more easy to obtain and can fit well into existing curricula. Even if they stray from the intended material, the exposure is at least a change for the students and whatever the lecturers say they are likely to reflect a desirable air of reality. Data processing people often perform internal training, make presentations, etc. and many have adequate skills for one or a few lectures. They are limited in what they can do, since their experience is usually rather narrow, but the following topics often work well:

- The history of data processing in company X (if sufficiently candid).
- Comparative analyses of products of general interest (the results of a selection process for a data base management package, for example)
- Management issues: The planning, control and administration of data processing in Company X. If the speaker is a senior manager, he can often speak effectively about what general management expects from data processing.
- The history of a single large software project.

1.2. Borrowing the Organization's Resources

Many user firms consider it in their interests to acquaint students at local institutions with their nature and needs, with an eye toward attracting the graduates. The firm may be willing to provide regular facility tours, regular background and indoctri-

nation lectures, or regular use of facilities such as self-teaching and audio-visual aids or even machine time. There is danger of irrelevance or brain-washing, but mutual employment self-interest often makes up for this.

Hardware and software vendors are also willing to be represented, subject to the same dangers. However, much useful information can be imparted by:

- A system engineer presenting the intricacies of his employer's system software offerings.
- A salesman describing the product line, contractual terms and the role of the vendor.
- A branch manager describing personnel needs, qualifications, career paths and internal training courses.

Local industries are usually willing to provide advice on curriculum development on a continuing basis. This can come from membership on a curriculum development committee, from individuals or from a training and/or personnel departments. There is a danger that little of use will be provided, but the communications that occur are almost sure to be worth the relatively little time involved.

An industrial counselor for students may be more useful than a curriculum counselor. Such a counselling function might be provided on a continuing basis by a local user firm. It could include lectures about the realities of the job market, starting salaries and career paths. It should also include discussions with individual students about their options. Almost any senior member of an organization's data processing staff could be of some use as a counsellor; in some cases, a permanent data processing personnel specialist can be found. Provision of an industrial counsellor could become a DPMA Chapter function.

Finally, there is the traditional industrial role of providing financial support: Research grants, professorships, fellowships, equipment or meeting/seminar support. (Compared to other fields, there is probably relatively little direct financial support of data processing educational activities by industry.) One of the most useful forms of such support involves a "bundle" of mutually useful activities with financial support, as at the MIS Research Center at the University of Minnesota. Another version of the "bundle" is multiple-sponsor support of research projects such as ISDOS and PL/C, with sponsors benefitting from interim results. Unfortunately, few institutions can support "bundles" this large.

2. Faculty and Students in Industry

2.1. Faculty

The paid consulting activities of faculty members are undoubtedly the most widespread form of interaction. They are not necessarily directed toward educational ends, however, since the user will only contract for consulting services he thinks he needs. If the educator will volunteer his time free or at a reduced rate more opportunities open up, such as:

- The development of case studies. It is time-consuming and difficult to invent good, detailed case studies or to develop them from a distance. By spending time within a user organization, however, a trained educator can develop excellent ones. Users will often cooperate subject to the right of reviewing the result, particularly if the user's identity is suppressed (which is usually possible, if desired).
- Surveying management practices. It takes only a few hours of interviewing a given user's people to learn the practices in use for planning, control and administration, and to get an idea of how well they work. A teacher who can convey such material from experience rather than abstractly is much more convincing.
- Research projects. In an area of current interest where user behavior is significant such as the evolution of the data base administrator function, a statistical survey of users is possible. Sometimes telephone calls will suffice, and users are generally very willing to cooperate with the researcher.

Formal fellowships in industry and in government agencies are also possible. Pennzoil, Arthur D. Little and the Civil Service Commission are three sponsors of such programs. They are usually for a summer (though sometimes for as much as one year). The sponsors usually match the faculty member's academic salary and try to provide him with work which both supports his research interest and helps the employing firm. It is sometimes difficult to find suitable work in a given firm, however. Possibly a clearinghouse for summer fellowships could be run by DPMA or some other body, matching user's advertised needs with the interests of teachers.

2.2 Students, Sometimes with Faculty

Numerous colleges and business schools have some form of "intern" program, under which students work for users part-time to gain experience. The principle is excellent but the administration is difficult: It is hard for the user to find meaningful work for a partly trained and inexperienced student. Successful programs of this type require careful advance work by the students' teachers to arrange suitable work.

A variation on the intern approach is the "working team," which meets specific user needs and also contributes income to both faculty and students, It involves making up a team of students, directed by faculty, to perform any substantial and separable body of work the user needs done. It could be:

- An audit of a finished system, involving user interviews.
- Conducting an evaluation of the prototype of an on-line system, involving extensive operating of terminals.
- Converting a group of programs to a new operating system or machine.
- Documenting an extensive system.
- Even keypunching, converting file media or other relatively menial tasks.

Such work will obviously not always be intellectually challenging, but it has the advantage of providing an income supplement for both students and faculty and it may be that any exposure to a user's real problems is good. The "working team" is attractive to the user, because the rate is low and because detailed supervision of the team is provided by the faculty members; the users' supervisors do not need to be continually involved.

2.3. In-House Training

Night and short courses for continuing education are offered by many colleges and universities, and fill an important need. Not every interested student can or will attend such courses, however, particularly if (like a computer operator) he has a rigid and demanding work schedule. Courses may be developed and taught on the user's site, involving assigned reading, correspondence exercises and a minimum of classroom time. Such courses are particularly effective for:

- The lower levels of workers, particularly computer operators who have an urgent need to upgrade and no easy way to do so.
- Brief, specialized topics (e.g. relational data bases) which can be taught to a group of specialists in a short time.

Such specialized courses can often be prepared jointly with the user's training or personnel staff, reducing the faculty member's preparation load.

3. Desirable Changes in Behavior

3.1. Faculty

To many academicians, research is to be performed in a laboratory or a library: The business school faculties seem (in general) to be the only ones that accept the user as a primary research area. A more general adoption of this attitude would seem desirable.

The computer science-oriented faculty thinks of the ACM and IEEE as their primary professional societies. This is good, but the local user contacts that can come from DPMA, ASM and other organizations are often richer. It would be well for educators to establish more contacts with these, particularly by attending local meetings.

Many educators are unused to the idea of initiating contacts with local businesses or government agencies, but they must be the initiators: It is clear that few users will come to them. It should not be onerous, because most users appreciate the desirability of cooperation and will at least be courteous.

3.2 Users

Publishing or otherwise disseminating the results of experience is generally not fashionable among users. Sometimes there are corporate constraints on doing so (though they can usually be lived with). The main problem is to establish in the users' minds the idea that publishing good material is beneficial to all. Policies of DPMA and other societies could be oriented to trying to change the present attitude.

Working with educators or teaching in local institutions is generally accepted as a desirable public service, but not many employers will allow much paid time for it. Recognition may be more important than time: The individual who helps the educators rarely receives much credit for doing so. In many cases this might be easily changed just by the manager stating his interest, and perhaps initiating the first contact between his staff and a local dean or faculty representative. In any case, a great deal of unexploited potential interest exists among user data process-

ing people, and a modest effort by educators to initiate interaction may bring substantial results.

This list of opportunities is intended to initiate discussion, and hopefully to initiate some specific actions. A thorough survey would surely find more types and variations of interaction. Perhaps such a survey, with wide dissemination of the results, would itself be a useful DPMA activity.

Comment [Withington]:

With regard to borrowing the organization's resources we might note some examples. I recently read about industry/-school involvement by professionals in a user organization who counsel the kids on what they should take, and where they should go. I think of this not so much a task for a single individual, but as a function that might be provided by a local DPMA chapter. An individual who works in data processing personnel might volunteer to be the job counselor for people on the local campuses.

Then, I move from this borrowing of individuals into the borrowing of resources. Most organizations are willing to—consider providing resources to help their local educational establishments. It's just that no one ever asks.

I heard the other day that an LTV affiliate is offering TSO time on off-shifts. They run their big machines on the off-shift for batch processing. It's up anyway and being operated and the terminals are sitting there. So they have given an explorer scout troop (that meets evenings) all the terminal time they can use on a big computer, which costs the company essentially nothing. And why not? Apparently this is a real turned-on group.

I would like to escalate one level in terms of support-such as support organized for a particular conference or meetings-like we had here earlier today. Gene Smith referred to conference at East Texas State University. This was a good one, partly because they hustled and established ongoing relationships with a group of large users in the vicinity who provided a group of interesting contributors (both the speakers and the audience of the conference), and money to make it a success in the various ways that money can help.

I have a feeling that if statistics could isolate data processing users and their relationship to the academic community, we'd find a lot less money contributed per size (some way) than a lot of the engineering-oriented disciplines do. In other words, users don't feed money back to the establishment. Perhaps they've never been asked, or if they have not effectively.

Comment [McKenney]:

Our program, in part, started because Charles Bowen of IBM kicked in some free fellowship money which allowed us to go out and buy students. And I think universities in general, when you go into a new field, compete in funding for the companies' dollars which are already well funneled into geology, petroleum and engineering. It's difficult to get fellowship funds for such a program.

Question:

How does that money get pried out of them in the beginning? What does that money get paid out of?

Answer [McKenney]:

Well, you start talking to Mike Samek and to Paul Strassmann and other people. They gave us money for the CBIS program. We got some other companies to kick in $2000 for a project. They began to have an association with the school and could see that they could influence the courses offered.

Comment [Withington]:

That leads me to my next topic; something that seems to have worked well in some cases is the structured, continuous feedback type of contribution to an on-going program. I'm thinking of examples like the MIS Research Center at Minnesota with a set of regular sponsors who get regular service as a result and the PL/C Project at Cornell. And, I wonder Dan Teichroew—would you like to say a word about how the sponsors contribute to and get benefit from the ISIS project?

Comment [Teichroew]:

The ISOS Project is one that is being conducted at the University of Michigan in the Department of Industrial Engineering. The primary objective is to give students some real, live, internshiptype experience while on the campus. And the money from the project goes to pay the students a research stipend which helps them get through school.

Now the practical experience comes from the fact that you have a large system, which is now installed in perhaps 20 or 25 installations. It's a large, interactive system, runs under most of the major operating systems; roughly 50,000 FORTRAN statements; usually a partition of about 320K when it runs under

TSO; roughly 10 million bytes for the whole system; has its own data management system of the network type; and it has the other features that fit in with what Dr. Mills just said. The language is stated in a formal grammar, the system is implemented as a finite state machine and all the rest of it. Roughly half the students are undergraduates and the other half are graduates. They get a stipend for working on this project. They are involved with all aspects of such a system. The first, and probably most important, is that they have to be concerned with whether users will use it—because if they don't get somebody to use it and pay for it, the money will dry up. When somebody comes to see it, the students are the ones who give the demonstration and try to get people to use it. They soon find out if you don't have good documentation, nobody will use it. (University software projects have a reputation of being undocumented).

So, we find the students are motivated to produce usable documentation. We get user documentation. And, as has been said here several times, user documentation is probably much harder to walk through than software. And they also get invoived in the installation. To get people to use it, they have to go out and install their system.

They are also involved in maintenance, b e c a u s e once they've installed it, they're the ones that answer the telephone. And when some irate user calls up and says, "Why doesn't this work," they have got to answer.

They also become very much concerned with things like performance. We had a case, for example, where somebody said, "We want it installed on this machine which has all kinds of available time on it, what with one shift on a 158—so don't worry about performance, just get it installed."

So it was installed and three months later they called up and said, "It's now taking roughly five minutes of CPU time under TSO just to do the commands." The students get involved in how do you modify the performance on this.

And, last but not least, they're also concerned with keeping ahead of the competition.

This kind of education is extremely costly. The number of students that you can educate is relatively limited. We probably had 20 to 25 students involved with it at any one time, and that's about maximum that we can have on such a budget.

I haven't said anything about what the system does. But it is, I believe, the state-of-the-art, the advancement of the art system in that it is dealing with the problems that systems departments now have—namely, how do you get your analyst to be a more productive person? I don't have time to say what "more" is, but I do have some discriptive material here of what that is.

Comment:

Could I just add a word to Dan's point about this being very expensive? I think society maintains the talent in various kinds of ways. I think that if you look into our industry you find that education is fantastically expensive, but we don't call it that—we call it other things like "debugging" and "getting systems to run." We call it "normal operations." But we're doing all kinds of education, and what Dan is pointing out is simply that very often the cost is hidden.

Comment [Withington]:

Exactly. I guess if the industry finds out what its total expenditures for every kind of education are, it would be startled.

I have one more category in this list of borrowing resources from users that could be very productive: Soliciting user support for an on-campus chapter of a professional society or perhaps an honor society. In Texas I encountered an honor society called Epsilon Delta Phi. They have about 10 chapters trying to make it grow. The idea is to recognize good grade point averages. These honor societies serve a purpose in the world in other disciplines, especially if the graduate still feels a tie and comes back to the meetings after he's working in the industry and talks to the kids. They are asking for DPMA recognition as sort of the affiliated h o n o r society. I have no direct opinion on that, but the principle seems good.

The point is, though, industry with very little effort could enrich such on-campus meetings very much. First of all by having people give talks to the students, answer their questions, provide ten bucks for an award, or maybe host a meal or give them meeting room if they've run out of room on the campus. Very little trouble, and enriching that function very much.

I feel, myself, that one of the despairing sort of comments that came out of the talk earlier is

"How do we find time to get all of these things into the students?" Well, one way is to capture some of their after-hours time at professional s o c i e t y chapter meetings to teach them some of the experience and other things that are not convenient to work into the formal curriculum.

Comment:

Can I mention that we've had a lot of success in the last year in working with the Discussion Science Club, and having people come in. And it's been a two-way street because there has been a lot of informal recruiting going on. Most people know that the way you recruit the best people is not to send your recruiters in, but to establish contacts. This has been very useful to companies throughout the area.

Comment [Withington]:

I've noticed that in symposia or chapter meetings, if there are students around, there seems to be something added. It's rare that a student can present finished work or experience in the way that the formal speakers do, but just having them ask questions or present a viewpoint—maybe they have done some little bit of work and can talk about it—and it seems to enrich the process even for the veterans who know it all.

Comment [Mills]:

In teaching, you discover you didn't know as much as you thought you did. But one of the things that industry doesn't take advantage of is getting their technical guys out to teaching—as you've been talking about here—because it does so much for the industry guy.

I can tell you f r o m personal experience (I've been teaching since 1957) I don't teach just for those students, I teach for myself, because there's nothing so healthy as facing 12 graduate students because they don't accept any garbage. So, I can say as a personal testimony that a lot of things that I've been doing I wouldn't have done it if I didn't have to f a c e those students every week. And what I teach at Hopkins one year becomes an IBM course the next year. Great benefits revert to the industry, to get people in there and just let their adrenalin run.

Comment [Withington]:

The only reason I got tired of writing is because I had to start teaching.

Comment:

Harlan, I'd like to pick up that point and emphasize that there are difficulties in putting people on campus for six months of the year. But these are advantages to the university. I agree with what you say.

Comment:

There are, of course, some substantial advantages for industry also—and I was picking up on the point that Harlan made. It's been mentioned a number of times here about how far industry is ahead of the universities but there are occasional instances where universities are ahead of industry. Sometimes it's quite useful for people who can manage to spend six months of the year on campus to learn the state-of-the-art in areas where universities are ahead of industry.

Plus, universities have a tradition of sabbaticals for their own faculty members—and some people think those are just boondoggles for them to go puttering around the world. But, in fact, as Gene Smith was saying this morning, particularly for the people who have been on campus, it is very important to get this retreading. The same thing applies to people in industry. Retreading is a chance to go back and study. It's difficult for the managerial people—I understand that—but industry should think more about getting people onto campus for the individuals' renewal and not so much for the company.

Comment [Smith]:

I'd just like to mention the program being conducted by Sears and AACSB. Dr. Barnes is the representative of AACSB. Their fellowship program is not computer oriented—but there's no reason why there couldn't be one that is computer oriented. In the AACSB Program, the Sears Foundation pays the recruiting and transportation costs for several (20 a year) faculty members to go to Washington and work with some governmental agency. Now, there's no reason why there couldn't be an interchange where a person from industry comes into the university. We're getting more liberal, I think, about accepting people without a doctorate. The academician, correspondingly, maybe doesn't take the place of the industry type, but certainly he could be doing work in industry and both people could benefit. I think we should think along those lines.

Comment [Withington]:

I have observed that there were some problems—matching the research desires and the interests of a faculty member to something which industry wants done. Can it really be useful? Is he worth the money that industry is asked to pay him for a period of

a summer, or a year, or whatever the term is?

Matching interests is a key problem in this academic fellowship in industry. And I wonder if this might be a specific function of a professional society in running a matching service in which industry can be solicited for "yes, we could use somebody in this area," which would be useful and be worth paying some money. A directory could be published to the faculty who would say, "Hey, wait, I'd better go off and see if I could get that one." At the local level, there's not enough population to make such an activity work.

Comment:

It's difficult to send anyone into any organization and get very much use out of them the first year—even an advanced, highly-trained faculty member. The reason, I think, for putting faculty members in conventional situations—and only large companies probably can afford this—is merely to get a different point of view into the organization, someone people can talk to, and bounce ideas off, and get interaction with, and not to look for direct, useful work from them.

Maybe I don't take issue, but it seems to me that a university person can come in and multiply, and reach the people in phenomenal ways. I think we're in a business where, for example, good design can cut implementation problems by a factor of two. I don't think we should view a visiting academician as a person just producing a whole lot of work as much as the people around him doing something different, and with visible benefit.

Comment:

Don't try and fill a spot, so to speak.

Comment [Withington]:

Cicumstances alter cases and individuals vary. Usually some of these things will be perceived as directly useful; many will not, except perhaps in an indirect way, i.e., the stimulus. But only if we have a broader universe brought together through some facilitating mechanism, the spectrum of differences will be worked out.

Comment:

We explored this with a couple of Ph.D. types, who will remain unnamed as we couldn't come to a match between what they thought they ought to be doing for the price and what we thought it was worth to us to do the job. We have not entered into one of these arrangements yet. But we made a real effort with a couple of people who approached us to try it. There's got to be a way, but we haven't been able to come up with it yet to satisfy both our management and the guy that wanted to come in. It's tough to describe the work situation we're going to enter into.

Comment [Withington]:

Let me turn to another somewhat related issue of the academic person using industry as a source of information. First, there is the case method such as Harvard has developed. But everybody uses examples, or case studies at some level, and they're kind of tricky to develop, as many of you know much better than I. You have to know enough to spend enough days on-site, interviewing people with different viewpoints before you can do it right. I would just like to observe that, subject to some sensitivity sometimes in which anonymity, might be requested. I can hardly imagine any of my consulting clients who wouldn't be willing to cooperate if anybody ever asked.

We have occasionally performed surveys, and it's amazing how people are willing to put up with the survey for half an hour on the phone in filling out a questionnaire—especially if it's for some kind of altruistic purpose. When I receive a questionnaire from some Ph.D. candidate who is writing about something, I return it. There's no reason this can't be carried to a pretty sophisticated level.

Comment:

Government policy is that anything that the Federal government owns is in the public domain, unless very stringent tests qualify it for treatment as classified or non-releasable. And there are very few businesses or professions that don't have Washington representatives. I think that faculty people interested in those professional areas could establish a wealth of information available for the asking, sometimes for the cost of reproduction—studies that took teams of professionals a couple of years to prepare.

Comment [Withington]:

We have tailored education to the idea that there exists a local community of users, or perhaps a single, very large client, who needs, as they all do, and all are doing—retraining, and upgrading. And, to my way of thinking, more of this is done in-house than needs to be. If you could attract an interested local academic establishment which would take what they do already—perhaps tailor it slightly with specialized

courses—above all, make it convenient for the user people, teaching outside working hours or teaching on the user premises, or whatever is needed to get to the people and minimize the time away from work this involves.

I'd like to put in a small plug here for one of my pets, the computer operator, who is being asked these days to handle a massively sophisticated complexity in these multi-mode, multipartition, multiprocessor systems. And whoever talks about training him or her in anything?

He can't get off the job any more than anyone else, but maybe he could take an hour off shift each day for two weeks to learn more about the intrinsics of the operating system.

Question:

We have a problem in that respect, whether it be the vendors or whether it be the industry itself. We've established a certain mystique sometime ago as to what the computer operator was to do and today they can make it more complex—yet simpler, to some extent. We in the government happen to be downgrading those jobs where the computer operator isn't any more skilled classification-wise than the key entry individual.

We're faced with a population of over 250,000 people in government, and they are obsolete, and we've gone through three-and-a-half (perhaps four) generations. But systems that are designed today are going to be designed by those people who were trained back in the old days and who have never kept up with the state of the hardware technology. They would design their old systems. And our problem is how do we get the educationalists together with the population and develop some plan of action that can upgrade these people?

Comment [Withington]:

Well, Bill Higgins was telling me earlier that in the Washington Area this has happened, up to a point. Would you like to speak to that Bill?

Comment [Higgins]:

Yes, in part, it is an interaction between government rules and what government wants to do to structure things. Sometimes they overstructure. Civil Service has certain job descriptions and definitions are frequently out-of-date. They're now working on those in the data processing field.

One of their studies recently indicated that in the data processing profession, the managers should budget 15 percent of the man hours of work force to maintenance of education. And just for the heck of it, I tested that against my own career. The largest educator in the world is probably the Department of Defense. We had to work out systems of work to transform a couple of million civilians into effective military workers. And, by golly, they've invested 16 percent of my career into going to formal school—and I hope they got paid back for it.

But with the universities, in particular, sometime after World War II, we found that in the expansion we had picked up a lot of high school graduates in both civil service and military. They had moved into middle management. Some of them we wanted to keep but we knew they couldn't hack it if they moved on up without more education.

We also had the motivational problem. A lot of these people were going to quit and go back to a university. So we combined these various problems and we found some universities that would teach after supper on our property. Today, most of the conference rooms in the Federal government in Washington (not just Defense) are classrooms after supper. The University of Maryland, George Washington University, Catholic University, University of Virginia give on-campus credit and they tailor courses to specific needs.

The business curriculum courses for University of Maryland, for example, include a course called "Military Logistics." It is one of the credit courses leading to their degree in Business Administration. The accreditation people have accepted it. On the west coast, we've got The University of Southern California and down in Texas we've got one or two universities. What I'm saying is that the institutional hurdles, both in government, certainly, and in your accrediting institutions, have already been broken down. And I was naive, I thought industry was already taking advantage of this.

Comment [Withington]:

No, I think it's a good deal lower than that. I think that's typical.

Comment [Higgins]:

Now, the way we work it, we pay 75 percent of the tuition for the student. He buys the books and pays the other 25 percent. For some reason, if something's completely free, some people don't respect it. It paid off. We combined that with our own internal schools. We combined it with sending people out full-time

for graduate study, underwriting the whole cost.

Now, this may be cheaper in the long run. Getting people trained in what industry needs may be a lot better than paying a high premium on all your entrants coming in with a baccalaureate or a master's degree and demanding to be paid accordingly. Take high school graduates or three-year certificate fellows, and then after they have been on board awhile, you take the ones you know are worth it and you pay for their schooling.

Comment [Withington]:

Here I have suggested ways of getting professors out from their ivory towers but I have not spoken of the student yet. We heard this morning about internships.

Even more so than with the faculty it is difficult to find an internship which is meaningful to the student's education process, and also any use at all other than just having them underfoot in industry. When Jim McKenney spoke of the program at Harvard, he indicated how much time was needed by faculty to get these set-up and organized to the benefit of both parties. Again, perhaps a professional society could play a role in finding internship opportunities and match these with need.

Babson College, has been doing something which I consider interesting—they offer a working team made up of supervisory faculty and students from the colleges who will go out and do things for money—which is often useful to students and faculty. And they'll do what needs doing, some of which could be quite relevant. I'm running down a quick list of the things they were thinking of in sort of inverse order of meaning to the education process. They could conduct a performance audit for finished systems. They could conduct and evaluate the test of a prototype on-line system, sometimes a complicated task requiring a lot of people to push buttons. Students could do that. A little duller, they could convert programs—you've got a bunch of JCL statements to move from one operating system to another, there's no reason why a gang of students under supervision couldn't do that. Document a system, a little practice there, if you don't do it in-house, try it from outside. Even keypunch, move files from one place to another. Students could do it. For m o n e y it makes it look more attractive. And I have a feeling, as I said before, that almost any kind of interaction—being on the premises, taking to the people—they've got to have some absorption of what the climate is like.

It's pretty hard for industry to employ the students directly. It takes supervision. But this idea of the team might make some sense.

Conclusion [Withington[:

Winding up for these things to happen, and the interaction process to improve, it needs work. From my vantage point, there are some behavior changes on both sides that would seem desirable. From the faculty point of view, the very idea that research could be done out there among users and in their offices instead of in libraries or in laboratories, which is sort of the traditional academic t h i n g, should be recognized. And I think even computer teachers are guilty of kind of thinking that way.

This idea simply needs to be more accepted, and they need to get out there on the phone with their questionnaires to business and take advantage of what is apparently a highly available resource.

To make their connections, I would think they should spend more of their time attending the local chapter meetings of DPMA, ASM or ACM.

Initiate the contact. Industry won't do it, never has and I guess never will, but they seem receptive, at least up to the point of being polite. They accept cooperation as a good thing to do. Picking up the phone and calling the people, making the contact—perhaps DPMA can act as a help in introducing people.

On the user side, one thing that bugs me a great deal and that is that users won't publish their good work. There's a combination of constraints apparently. Some corporations say, "Why should I give away something I paid money for?" Or maybe it's a trade secret, or no brownie points for the guy who does it. The journals and both professional societies and magazines in the field are always looking for good material, and some data processing people can write. I really don't know why there isn't more of this, because there certainly is material there.

Encourage individuals in the data processing organization to work with local schools. I think there's a lot of willingness in principle. They might even enjoy it. But management, perhaps, is worried about taking too much time from their jobs, but more than that, management doesn't seem to care. It might just be

enough for a data processing manager if he said, "Hey, fellas, I would consider it a good thing if you can do this and if you want to do it without interferring with your work." And maybe then they'll wander off and do it.

Get together with the training department of a user to learn what they need, what can you get from them. Think about this business of freeing people to retread, to go back for a sabbbatical from industry to academic, and pay them for doing it because it might be cheaper (as Colonel Higgins indicated, especially if it can be done out on site) than to put so much money at the front end.

Finally, I can see the role of the professional society as a very powerful and important facilitator, sort of a switchyard for experiences from one side to the other—attracting the people, publishing lists of available opportunities. I think we can leave today with DPMA itself having a few actions it can take as well as a broader view for all of us.

Section Four: Workshop Discussion Groups

The fourth section of the workshop included discussion groups followed with reports of each session by a discussion leader. There were six questions which the groups addressed. They included:

1. What specific types of action programs should be taken to better prepare information systems graduates for careers in industry and government?
2. What specific types of action programs should be taken to improve and maintain the skills of current practitioners?
3. How can these programs be implemented?
   A. Role of industry?
   B. Role of academia?
   C. Others?
4. How can such programs be funded?
5. How can the findings and recommendations of this workshop best be disseminated?
6. What role do each of you intend to play?

The discussion leaders were: Max Hopper, assistant vice president, SABRE DP & Communication Services, Tulsa, OK; Paul A. Strassmann, director, administration and information systems and general manager, information services division, Xerox Corp., Stamford, CT; D.R. Wolf, general manager information and computer service, Shell Oil Co., Houston, TX; Felix Kaufman, national director, management consulting services, Cooper & Lybrand, New York, NY, and Richard G. Canning, Canning Publications, Inc., Vista, CA.

The reports of the group leaders plus the discussion which followed is included in this section.

Max Hopper

We found there already exists considerable interaction, at least as seen by members of our group between industry and government, on one hand, and the institutions represented—four from industry and one from government. In trying to assess what we had done or been involved in, most of the things that were discussed this morning had been covered.

For example, intern programs. Three or four of us had intern programs. At least two or three had co-op programs. I don't think there were any who did not sponsor educational programs to assist folks in on-going or continuing education, either at the undergraduate or graduate level.

Specialized training was offered by the industrial institutions in cooperation with educational institutions in some cases, either on the premises of the company or for the local community as a whole. Some of these, particularly one done with a DPMA chapter down in Bloomington, IL, led to a much broader community relationship. And there were those where teaching was done on the university campus on a part-time basis; none were on a full-time basis.

The point I'm making is that, from the industrial side of things, it seems that all the activities or the potential activities that were discussed this morning were covered in one way or another with participants in our group.

Both of the educational institutions that were represented had done quite a bit of surveying to find out what skills industry or business needed in terms of putting together their curriculum and what they were training or teaching their undergraduates. And both of them, in contrast to some of the comments that were made this morning, are training at the undergraduate level as opposed to the graduate level in terms of trying to put something together for the industrial community.

We then asked ourselves what programs we were seeking educational institutions to provide for industry in data processing education. There were mixed feelings, some of them strong. For example, that the two-year colleges did provide an excellent caliber of employees at a certain level that could be productive a fairly short time after some internal training.

No one in our group felt that educational institutions should be totally responsible for providing an individual that could walk in the door and pick up a coding pad or start to do analysis. Every one of them felt that they had some responsibility for training that individual. In fact, I think State Farm said they wouldn't even allow them to touch anything for six months, until they had gone through internal training.

I got the general impression that most of us felt that graduate-level new employees sometimes did not exhibit much humility. Someone has to tell Master's graduates (and perhaps others) that it does take time in a business enterprise to reach a management level position. We are trying to stress this in our internal training.

Communications skills came up as really important.

Several of us commented on the approach that Harlan (Mills)

mentioned earlier this morning. Some of us would prefer it perhaps a little less theoretical; one gentlemen called it the "scientific method," another commented on it as a "problem-solving" approach. I think it boils down to taking a very logical approach to solving business problems as proposed to prematurely jumping into coding.

We do feel there's a role to be played here between the business and the academic world in terms of preparing programmers and others for data processing installations. We have to work together. Specific action programs were fairly well covered this morning. The question is: Can we make them work better?

Although the role of industry, we feel, is to put pressure on the academic world to train in things that business needs, some of us felt that the academic world should turn out some generalists, that we couldn't have only specialists.

We felt that the professional societies could play a very important role here as a catalyst, as the Education Foundation is doing today in stirring the pot; and perhaps also in disseminating educational opportunities, or combinations of educational and industrial opportunities and in providing information on what can be done and what is done by others.

We felt the marketplace would probably wind up dictating what would happen in any event.

Paul Strassmann

We took a narrower view of our assignment—looking at what could be prescribed to the Education Foundation as an immediate action program. The Foundation would price out budgets and assess feasability. And we came up with six specific actionable recommendations.

Recommendation #1

We recommend that the Foundation set up a clearinghouse of industry personnel available to educational institutions. By "clearinghouse", we mean classes of commitments that industry people are willing to make either to local lectureship arrangements or, in those cases where industry wishes to make available individuals for assignments to the universities or research institutes, for extended lengths of stay either for refreshing the technical equity of industry people or for the purpose of acting for industry in the same role as academia enjoys in case of sabaticals. The clearinghouse would be a fairly low cost operation in our opinion and eminently desirable. It could operate with suitable circulars and reports. It would perform an extremely valuable function, one which today is not being fulfilled anywhere in industry.

Recommendation #2

We recommend that the Foundation operate a registry of opportunities for academia/industry exchanges, because it's quite clear that industry does not know, for instance, which institutions have graduate students either available for research projects or Ph.D. thesis arrangements, such as those mentioned this morning by Jim McKenney of Harvard.

Similarly, there will be people from academia who want to spend a sabatical in industry and consequently, a registry of these opportunities would be desirable. This is again a gap that needs filling, and I think the Education Foundation is more suited to perform such a service than anybody else available.

Recommendation #3

The Foundation should publish an annual directory of educational resources. As the industry becomes more pervasive and more articulated, the variety of programs, such as Dan Teichroew's, should be highlighted for visibility. The Education Foundation could provide the proper mechanism to make these opportunities and programs visible. Such a directory could also lead to commenting on the curricula and how the various activities of various universities fit into some kind of discernable scheme. This directory has evolutionary opportunities in terms of enhancements that could be added to it.

Recommendation #4

This would require investment by the Foundation in research. It is apparent to all of us that there is far from a unanimous view of what the future will bring. We are talking about millions of computers out there, and about the end of the life cycle of the educational investments that we are making today. Remember that the people who are coming out of graduate school today will be retiring in the year 2020, '25 and '30. Therefore, we are talking about long term investments. The Education Foundation could perform an inestimatable service to the industry by using the techniques of dynamic input/output models to project the most likely shifts in human resource

requirements as time goes on. The input/output models would have (as rows) the categories of skills necessary at a given stage of industry development (most likely in 5-year increments) and (as columns) the educational inputs necessary to satisfy the demands of the environment. And as our economy is tending more and more to be human skills-oriented, the investment in this tool, and in making the results of that research available to industry, could have tremendous leverage in orienting the industry (as well as the educational institutions) toward the kind of program investments needed.

Whether the investment should be 15 percent of payroll, as was quoted this morning, or three percent of payroll would come out through an input/output modeling exercise. This likely will be the most costly of the Foundation's programs. It's a long-term program, but there is nobody else even on the horizon to perform this function for the industry.

Recommendation #5

The Foundation should appoint industry/academia steering committees to guide the implementation of the above recommendations. Each program is of sufficient scope and size that the Foundation would need some sort of an organizational structure to keep them headed in the right direction. There should be some way to attest to the industry and to the academic world that, in fact, their interests and policies were properly considered. Steering committees of four to six professionals per program may be perhaps the optimal size to guide the dissemination of these results.

Recommendation #6

Funding for these programs should be by subscription. The Educational Foundation, on a program by the program basis, should solicit sponsors; in some cases the (I/O model, for example) the Foundation should invest in a business proposal, which it would then circulate. Based on these proposals. it should determine a reasonable sponsorship fee per program; and then, based on areas of interest, solicit sponsors. For all the programs we have recommended here, we are talking about an investment of no more than $50,000 to $100,000 per annum.

Comment:

This meeting points out the lack of communication generally in the industry and among different groups within it. For example, ACM already publishes a directory of curricula in educational programs. AFIPS, under Bruce Gilchrist, has done long term planning studies on the need for people in industry at various skill levels. And so the seeds of such programs are around, waiting to be picked up and funded. In the cirricula development area, Dan Teichroew and others have done work for ACM. It's frustrating that more people are not aware of these. Somehow, this information has got to be coordinated, before we go off and start spinning our wheels again.

Comment:

I'd like to strongly support Recommendation #3, the directory of educational resources. In working on this project this year and on my related recruitment effort, I was absolutely staggered at the variety of educational offerings that the universities offered. To take a case in point, Dan Teichroew from the University of Michigan is associated with one of four programs that we have an interest in there. Now, in relation to the ACM publication, I think there were 21 schools listed but there was no glimmer that would have led us to suspect, for example, that Michigan has four programs. Michigan is not unusual; this is a very common thing on the campuses we visit. Recently I said to a colleague, "We could have stayed for two or three days, because there are two or three different schools here that are turning out different curricula and each one has a different training objective and a different orientation." Now, from our standpoint as recruiters, it is very difficult to sort them out and have some idea as to what the offerings are. It must be impossible from the standpoint of the kids trying to select which school they should go to.

Comment [Strassmann]:

I am aware of Bruce Gilchrist's work for AFIPS, and also of the ACM reference material. But we are recommending a much more comprehensive effort here, and with much greater reference, retrieval and indexing capabilities than are currently available. We are talking about scaling up by an order of magnitude.

Comment:

There are bits and pieces of efforts all around. In fact, there is a substantial base to draw on in some of these areas. We know there are several very useful curriculum recommendations that have been made by Dan's (ACM) committee. I feel though, that a lot of those recommendations are gathering dust, and that's most unfortunate.

Comment:
I don't have much use for curriculum recommendations *per se*. What's out there in the place right now? I share the frustrations about what's going on on campus. I think it should be coming from the business side of the house rather than from ACM. What's there? Give us some input!

Comment:
I'm really saying the same thing—not the idea of making curriculum recommendations, or even compiling recommendations made by others so much as just giving us a comprehensive directory of what kind of things are taught by which schools.

D.R. Wolf

For the most part, we are not looking for someone highly trained, but rather we are looking for someone highly educated. We distinguished in our group between education and training. We considered whether there could be a contribution from the academic world to our following training and pretty much came to the conclusion that the academic world could offer considerable help in the area of continuing education, but probably isn't all that well qualified in the areas of specific follow-up training. We felt the area of continuing education, which most of you are involved in, was of importance to us.

We also felt it was difficult in this area to talk about action items but we agreed that the industrial world had not articulated its needs completely to today's students. I think the faculty probably has a very good understanding of how we feel. But we are getting students who are over-ambitious or their ambitions are higher than we realized. We felt that perhaps we had an obligation to thoroughly articulate what's really going to happen to the student. What are our real needs? What are the career paths? What are the opportunities for continuing education? We also discovered that everyone is doing something in the area of interaction—intern programs, civic-supported grants, or what have you. And we felt that there should be a mechanism by which someone coordinates.

We then dropped off into the fragmentation of professional societies again. Some of us have been involved in the fragmentation issue for longer than we care to acknowledge. But there does seem to be a need for somebody to step above the more emotional and, in many cases, not very productive discussion and generate some sort of compilation of what's available. I don't know about all of you, but I would imagine nearly every one of us comes away from here hearing of at least one or two specifics that we really hadn't heard of before, whether it be a full exchange program such as the Sears program or the Harvard/Celanese/-Xerox interrelationship.

I think we felt there was a need for some methodology by which we could better understand and be aware of those things that are occuring, whether they be internship programs or specific work programs. I would imagine that half the industries represented in this room are working with Explorer Scouts. But how many of us are involving each other in planning sessions? How many of us are sitting down and talking about software design? Those are the things that seem to me to be missing. I have a feeling that if we're not careful, we can fall right back into the old problem that we've been in for 15 years in this business, and that is, can someone step above and bring about the compilations we need without stepping on anybody's toes?

We did get into the area of support and interaction and we saw support as not being necessarily one-way but rather as part of interaction.

We talked about financial support, and in final analysis, financial support is going to be required. But I do think that many areas of financial support are going to require some semblance of cost/benefit relationship—at least a business ap-

proach to requesting financial support. The justification area is one that I don't think we can leave untouched if we propose a program that requires financial support.

It is true that it's hard from the "mother, God and country" standpoint to say anything bad about most of the suggestions we have come up with; but when we get back home most of us are going to sit there and look at our computing budgets and we're going to have a little difficulty. And yet I think we should support these programs. But I would hope most of them are addressable in terms of cost benefit relationships.

Then we touched on the fragmentation of professional societies, and I don't mean to play that issue up, other than to point to it as an impediment to our effectively developing some sort of a clearinghouse approach. And we too, saw a clearinghouse approach as a means of information exchange.

Most of us don't publish and don't write because we don't have the time or the ambition. Last night, I was lectured on doing that several times, but we don't do much publishing in our business.

I do think that there are ways to encourage greater cooperation and ways to profit from the experience of others who have been successful at it. It would help, I think, to tell us of the things others have done, things we have put off doing.

We had some very emotional arguments in our group. What may have come out of our group was a greater respect for each other—even though we don't agree on everything.

The coordination opportunities, so well described by Group 2, the awareness aspect and the need for generalists as touched on by Group 1, would be of interest.

Felix Kaufman

I would describe our session as spirited, intelligent and chaotic. And I think, inserting my own opinion, it reflected the fact that it's pretty tough dealing with this issue and at this point trying to get to recommendations for action. So we kept talking about the issues, and we'd be at it for the next few hours if the session hadn't ended.

One important conclusion reached was that we now know enough about how you build and manage systems—and, emphasizing the word "practice," that we have a sensation about what the structure of the body of practice in the field now is, and that it would be useful to begin efforts to codify this; because if we can accomplish something in this area, we could provide the academic community with a structure to cope with. Then, perhaps, their efforts could be contained within that structure, instead of being open-ended.

It was also observed—I don't think with the intention of being a cop-out for the academic interests here—that, as they see life these days as suppliers, they don't have any trouble seeing the supply higher. And so that at the moment, perhaps, the problems that are perceived in the marketplace are user problems. Those of us here representing industry and government are the ones who are saying, "Well, you guys are not having any trouble turning out a lot of people in this area and we're hiring them all. But we're not exactly satisfied with what we get."

And this led to the conclusion that there ought to be an evaluation of what users really think they want. And, it was added, that it is treacherous to ask only the users, because there may be a gap between what they tell academia they need, and what they in fact require for the way they actually use the resources that they finally obtain. So we agreed that there ought to be a survey of user needs on a "for-real" basis.

I related that because of Paul Strassmann's suggestion of an input/output model. There were some collateral observations in our group. First, the points I mentioned: Do we know the demand? What kind of demand is there? Are we allocating the resources that are provided in quite the right way? There is an interface between our observation about clarifying user needs and the observation Paul made about the usefulness of a personnel input/output model.

We thought that if it were possible to clarify user needs, it would relate directly to influencing programs and, of course, ultimately courses.

There was some feeling in our group that some trends in the data processing field are so intense at the moment—for example, structured programming and data base management systems—that curricula, anchored to programs as specific as these, would produce people of greater value to industry at the present time, although there were notes of caution and qualification inserted in the discussion of those points.

To further illustrate the point about trying to do a better job of defining the different career ladders that we may be concerned with, we observed that as we move ahead, it may make more sense than we may have thought

it did to have systems factories, so to speak that would develop our systems rather than to rely as heavily as we now do on the user doing that. And whether that is the case or not, that this is a desirable tendency. Some of us feeling that perhaps the people who are now being trained are better oriented to the systems production process than they are to the user organization.

To make that clearer, the point was amplified by saying that perhaps many of the people now taking majors in the computer field should be counseled to go with a high-grade software house if they had that career option rather than with a user organization, this being more compatible with the kind of training they're getting.

We deplored the influence of existing accreditation processes in the academic community. These seem to force greater and greater emphasis on accepted programs and result in using up the possible capacity for the kinds of course material that we see as becoming important and needing room (there isn't room in the college program because of the status-quo accreditation processes).

We didn't have a very good plan, though, for trying to bear down on that except to make our feelings known.

There was pretty much of a concensus that more should be done in the area of cooperative programs—that is, industry cooperating during the semester or quarter on-campus and then the semester or quarter off-campus in a work program—as being particularly adaptable and desirable in this field.

Toward the end, we tried to have a meaningful discussion of what is needed or what it would take to maintain faculty awareness in this fast-moving field. And aside from the concern as to how this could be accomplished, we really didn't have sufficient time to deal with that particular concern.

I'll conclude by giving you one opinion of my own. I have a general feeling that from the user side—and I consider myself one—we're too tough on the academic community. If they try to adapt themselves to what they would perceive if we all could articulate the full complement of user-end requirements, they'd pretty much go crazy because it's kind of an open ended demand. Everybody has a slightly different need. Now when you pile this all together, it's tough to cope with.

I also think that the more the academic community tries to do this, the less time there is left for the basic skills that a series of speakers laid on so heavily this morning—a point of view with which I fully concur. So we must keep in mind that no matter how good a job academia does we still have to invest a great deal of training in people from the campus to adapt them to our environments.

Comment:
Two years ago I attended a meeting chaired by Dick Canning on the production of a software engineering handbook, and I haven't heard much of it since. I wonder if Dick could tell us what the status is.

Answer [Canning]:
Yes, that was a joint effort between the National Bureau of Standards and ACM to indicate what the table of contents might be for a software engineering handbook. Shortly after the meeting, all hell broke loose at the Bureau of Standards and everything stopped while some big hassle from the Secretary of Commerce had to be settled. The report was published probably a year after we had our meeting, and NBS has, subsequent to that, appointed an editor who is trying to get the chapters written for this handbook. I have heard no further progress reports on it. It may still be alive, but I haven't any indication that it is particularly healthy.

Comment [Teichroew]:
Felix mentioned the accreditation problem, and perhaps I should note that those of us in the computer science department—and I guess I'm the only one here who is—cannot blame any of our malfeasances or other faults on accreditation, because in computer science, as opposed to schools or business or engineering, there is no accreditation of any kind and none on the horizon.

Comment [Mills]:
I should speak up on that handbook business, because I've become one of its editors. We had a meeting and decided we didn't have enough horsepower to edit it and so I think the Bureau is now trying to put together enough horsepower to do so. They're getting a slate of subtopical area pages. I don't know when or if anything is going to be published.

Comment [Glaser]:
I attended the meeting at UC Irvine on software engineering (referred to earlier). Those attendees who are trying to teach

software engineering openly acknowledged their dilemma and asked the other attendees to "tell us what software engineering is because we don't know what to teach."

Perhaps some of you here today could help them by describing what software engineering means to you and what you'd like them to teach. The academic community is asking for help and, if the Irvine meeting is any indicator of the general state of affairs, it is searching for a practical working definition of software engineering so that it can identify the subjects it ought to teach.

On a different subject, AFIPS launched an ambitious program about five years ago to produce a series of what were called "Systems Review Manuals." The initial objective was to come up with certain standards or criteria by which you could evaluate a system you had built (or a practice you followed) against some norm; presumably, this would help you know whether you'd done a good job or not. After some delay, AFIPS did produce a manual on security that has been very well received (four or five-thousand copies have been distributed). But it was a horrendous task to get it done. We gave up on volunteers and wound up paying an editor to finish the job.

The question then came up, what will the next topic be? And a very popular answer was software engineering. But despite the fact that there was general agreement that a software engineering manual would be a good one to write, the matter immediately fell into the tnicket of jurisdictional disputes among the various members of AFIPS, and there it continues to gather dust.

Comment [Ralston]:

AFIPS now has a proposal before it—proposal is a slightly strong word—that would serve to continue the review manual program; it will be discussed in a week or so by the AFIPS Executive Committee. The point that I would make is that the person who proposed this said that in the long run, this activity might well be as large as the sponsorship of the National Computer Conference in terms of resources required.

Richard Canning

We spent a little bit of time discussing how to better prepare information system graduates for careers. What did it mean? I asked the group whether graduates were to be considered entry-level as far as industry is concerned. Howard Morgan felt, "No," that those people who had Master's degrees shouldn't be considered entry-level.

Others felt, "Yes," they are entry-level. They should know what the backroom activities are. It is important to be aware of what goes on in a day-to-day business.

But there was concensus on the need for a better balance in the educational program for information science graduates, for example, between technical areas, operations research and management and social awareness. Surveys, too, are needed, such as the one Mayford Roark described this morning, to feel out what is asked for. This has to be tempered by what was mentioned a moment ago—where do the companies actually put their money? Who are they hiring and how are they allocating the people they have? Surveys of this nature would be helpful.

So, a better balance in the curriculum is needed coupled with some form of field experience. This could be an internship program that would get the students out into the field to explore what's going on in business, and to learn a little bit about what's going on in the so-called back room.

A related point here about the curricula/inter-departmental degree programs that were mentioned favorably, where more than one department—computer science, engineering, or information science—combine to give the breadth of education that's needed.

How about the people who are already working—the practitioners? One model mentioned was the AT&T program that was conducted for the middle management of the Bell System—and out of which grew an M.B.A. program at the Wharton School. This same concept could be turned around the other way—if a school developed a pretty good M.B.A. or Master's program, portions of it might then be offered to middle management in some form or another.

One other point on how to upgrade current practitioners: Bring key people, well-known in certain aspects of our field, into the company on a regular basis. A different person every two weeks or every month could talk to groups of people for half a day, and hold discussions with people in the company during the other half, just to make them aware of what is going on.

We discussed one other aspect of training that deals with upgrading people on the individual level. Management does not often think of an educational program affecting large groups; in most instances, staff upgrading is planned on an individual-by-individual basis. But larger questions need answering: What are their career paths? What are their opportunities? And what training should each one be exposed to?

What can be done by the Education Foundation? What sort of actions could it take? First, stimulate the universities to teach what is needed. A maintenance type of survey was

discussed earlier. That would be a most interesting way to identify general types of courses, and some statistics based on a sample that would give a feel for what systems managers want. It's not uncommon for me to hear people in the business world complain bitterly that the universities are not turning out people that they can use. The right kind of survey would give us some numbers that could be attached to specific subject areas.

If universities are to teach what is needed, they're going to need some support. What types of support are needed? Industry might very well send more people to the university programs, and hire more graduates of such programs. It could provide grants via support for education programs; support faculty training and adjunct professors; bring professors in as consultants; perhaps to work on task forces, perhaps to come in once a week or for a period of time as a resource to be used as needed.

Dissemination of the results of this workshop should be made to the deans of all schools of business.

The Foundation could also send this material to the local chapters of professional societies. They, in turn, could go knock on the dean's door and say, "Here's something good we think you ought to take a look at."

High-level industry endorsement of something like this is needed. Curriculum '74 came from a group of dedicated people representing academic institutions who worked for a good amount of time. But I don't believe that any high-level industry group said, "This is the kind of curriculum we would like to see." For anything that comes out of the Foundation, it would be very helpful to have the endorsement of a group representing major industry.

What can be done to emphasize the importance of off-site short (two-week) courses? The summer institute at the University of California at Santa Cruz seems very successful. It lasts about six weeks and consists of a series of two-week courses. Qualified data processing people come for some specific subject and they expect to go away with a capability that they didn't have before. Every course involves lecture and lab. And these are reasonably advanced—one is in compiler writing. Application programmers who haven't the vaguest idea of how a compiler works have come and in fact, have gone away with that capability. I know of one case at the Department of Labor, where a statistician took the course, went home, designed a language much more suitable for handling the statistics he was dealing with, and wrote the compiler—he and one of his co-workers—that is not only in use in his department, but also is being used in a number of other installations.

It seems to me that things like this could be done; and if industry were to support it, we'd begin to see much better interaction between academia and industry.

The final suggestion that was brought up was that you talk to your company "Foundation"—some companies have foundations, others have individuals who serve in this capacity or the people in charge of corporate support for education. They seem to be a key group with which the Education Foundation should communicate.

A little bit earlier I said things ought to be communicated to deans; the Education Foundation ought to develop a mailing list of people in industry and government who are in charge of corporate support for education and another list of the deans of the business schools with whom you want to communicate. So, disseminate—use the good old U.S. Mail to help you get information from the Foundation out to where it's going to do some good.

Section Five : American Association of Collegiate Schools of Business

Dr. B. L. Barnes, dean, College of Business, University of Iowa, Iowa City, IA discussed the American Association of Collegiate Schools of Business. His presentation is covered in this section.

Dr. B. L. Barnes

I have enjoyed the meeting today, as an outsider sitting in. I find that most problems are common, at least they now have more commonality than yesteryear's.

First, I would like to make some comment about AACSB, because it seems to me there's gross misunderstanding. I might as well start from the beginning. AACSB was organized as the American Association of Collegiate Schools of Business in 1916. In 1919, it promulgated its first set of standards. March 1, 1975, a new charter was drawn up and bylaws; and while we kept the same acronym, the organization became the American Assembly of Collegiate Schools of Business.

Membership is voluntary. It's composed of 176 schools that have accredited programs; approximately 250 or 300 schools that have programs that are not accredited—this is out of a total of some 800-850 colleges and universities around the country that have the programs. It also has 77 members from industry, government agencies and organizations. And, in addition, it has 32 international members. The international members are agencies, organizations and some schools.

Standards. What are they? How did they come about? First, every standard and every amendment to standards is approved by the accreditation council, that is, the accredited schools. I should have told you that to enter structure of AACSB, originally the membership was synonymous with accreditation; that is, all members of the American Association of Collegiate Schools of Business were accredited members. As we brought in the non-accredited schools and the business/government/agency membership, then the accreditation council became the substructure that is solely responsible for accreditation of schools of business, recognized as such by HEW as the official accrediting agency.

So, the standards then come about from the member schools themselves. They're not dictated by some authority, they're not handed down on tablets; they simply are the concensus of the member accredited schools of the set of standards that we feel we should hold to.

As you get into the standards, I think most of you would be more appalled by the laxness of the standards rather than the constraining effect of the rigidity.

Let me run through the standards with you briefly. There are eight categories of standards. The first one is the ultimate objectives. From experience it's been found out that like businesses that don't have objectives, schools aren't going any place necessarily unless they determine in advance what the heck it is that they're trying to be. We do require schools to spell out their objectives, and we don't require all the objectives to be the same. There is a heterogeneity in schools of business that you wouldn't believe. Today, for instance, as it is when we talk in faculty meetings or most any other place, the term "school of business" is used as if there's a homogeneity there that simply doesn't exist.

I used to say that schools of business were similar in name only. But that isn't even the

truth. Among the 176 accredited schools at undergraduate level, they are known by 69 different names. Now that doesn't allude to the Sloan School or the Wharton School—this is School of Business, School of Business Administration, School of Business and Economics, School of Management, and so forth—69 different names.

At the graduate level, among the graduate schools of business (of which about 79 of us are accredited), even those programs are known by 17 different names. So the first thing than, you don't want to make the mistake of assuming any sort of uniformity in schools of business. Quality-wise, there is a wide spectrum of quality. It ranges from the prestigious schools, who've been the leaders for years—Harvard, Stanford, Chicago—all the way down to schools of business that we, in the profession, are embarrassed by.

Among the accredited schools, then, we do have standards. So the first is the objectives.

The second standard deals with admission of students—the quality of the students that are admitted into the program. And it has come about for necessity, because schools of business are not autonomous within universities, and we do have universities that are numbers-conscious. We have boards of trustees and we have presidents that are pushing to get every enrollment in there that you can get. So we do set standards on admission, both at the undergraduate level and the graduate level.

The third standard deals with personnel—faculty, academic staff, non-academic staff. The fourth deals with the curriculum. I'll talk more completely about personnel and curriculum in just a minute. The fifth with libraries; the sixth with financial resources and facilities; seventh with innovation and technology—it is a requirement that a school demonstrate innovativeness and technology update; and finally, the eighth, adherence to standards.

Now, the misunderstandings that seem to be cropping up today. First, let's look at the faculty. The pertinent requirements of the faculty standard: First, seventy-five percent of the faculty must be full-time. Now what does that mean? And is it really a constraining feature? I have approximately one hundred FTE in my college at the University of Iowa. That means that I have to have 75 of those FTE full-time. That gives me a great deal of latitude to have any visitors from industry or from business from any other source.

We also have a requirement that says 80 percent of the faculty must hold appropriate degrees at the level of Master's, Ph.D., or other appropriate designations such as J.D., M.D., RR or CPA, etc.

We also have a requirement that says 40 percent of the full-time faculty who teach at undergraduate level must hold an appropriate doctorate. Forty percent. That isn't much. I can get by with about 25 on my staff. I have 58 out of 72. At the graduate level, 75 percent of the faculty who teach at the graduate level—full-time faculty—must hold an apropriate doctorate. So what is the constraint in that? It's a *low* standard, in my opinion.

Curriculum? Let's look at what is required here. There are many myths about AACSB, and I think it's because too often it's been used as a crutch, either for things that a faculty didn't want to do, or an administration didn't want to do; and sometimes it's been used as a coercive myth for things they wanted to do and were looking for reasons why. For instance, you go back twenty years, departments of business education were in schools of business. Most of them are outside the schools of business now and I have seen reports to boards of regents that said because AACSB was "putting pressure on them," the AACSB has never taken a position on such things as teacher education.

Take 126 hours—that's a typical undergraduate program—what does AACSB require in curriculum? First, we require that 40 to 60 percent of the work be in business administration, which means as many as 75 hours of the work can be in business administration. It also requires that 40 to 60 percent not be in business administration.

So, out of 126 hours, 75 hours of it can be in business administration. My personal conviction is that that's too much and I won't let it go that far at the University of Iowa.

What specific courses do they require in that 75 hours? The answer is, none. AACSB's standards do require that a certain percentage of the work be devoted to covering five areas known as the "Common Body of Knowledge." It does not designate how much time is to be given to each subject. It does not specify that any specific course has to be

offered. It does not require that accounting majors, marketing majors, finance majors, all meet these Common Body requirements the same way, but it says that approximately—that's 25 percent or more—of the four—year program must be committed to this Common Body. That's interpreted in the standards to mean 30 hours.

So you have 75 hours potential in business courses; you have 30 hours that must be committed to five subject matter areas. That leaves 25 hours for a school to do what it jolly well pleases. I can't conceive of a program—whether it be accounting, finance, marketing, information systems, or data processing—that can't be accomplished in much less in that 25 hours that's left.

So. I guess that I would categorically deny that AACSB has provided a constraint on the development of EDP. To the contrary. It has required that the area be brought into the curriculum, and specifically has encouraged the adaptation of the computer into the curriculum.

I guess all I'll say is that when you go back to the charter, the purpose of AACSB is to foster the development of schools of business, not to constrain them. And I have been connected with it quite some time, and I haven't seen the constraint.

Conclusion

Comment [Glaser]:

I want to say a word about the Education Foundation before we begin to discuss the suggestions which came up in the working groups. We exist at the moment through the generosity of DPMA, which has provided us with the start-up funds. These funds are used primarily for meetings of the Foundation's regents, which take place about every six weeks. So far we have no other sources of income. We have positioned ourselves with the taxing authorities as a tax-exempt organization so that funds given to the Foundation are deductible to the donor.

In order to respond to the ideas that have come up today, we will not only have to put together programs, but we will also have to sell them to somebody who will be willing to fund them, whether a private foundation or an individual company. Paul's group suggested the idea of subscriptions; we will look into that to see if we can't come up with a scheme that makes sense.

How many of you work for companies that are now involved in subscription-like support of a body like this? There is a precedent for that. Good! I know that the Diebold Group and Arthur D. Little, and some others have subscriptions of one sort or another.

Paul's group also suggested an industry subcommittee. There's no question that, to the extent that we can formalize some of these programs and focus on them, we're going to need help from industry subcommittees who will guide what we do. We are a relatively small body and have limited resources ourselves; all of us are otherwise employed-and we will need guidance on a lot of these proposed prospects where the Board's experience just isn't adequate to take them on.

There were several calls for directories of one sort of another. One such directory would include academic programs, course contents, the objectives of those programs, etc. I take it that there is no such catalog now that is satisfactory—is that true?

Comment:

There are several catalogs, but the "satisfactory" requirement may have ruled them out.

Comment [Glaser]:

That idea came out of Max Hopper's group. Max, are you familiar with what's available now and do you consider it unsatisfactory? Or can you speak for your group on that point?

Comment [Hopper]:

I'm not sure I can speak fully for the group, but I'm sure we are not totally familiar with what is available.

Comment [Glaser]:

The chances are that Dan Cougar's newsletter is not in the hands of most industrial sites nor was it intended for that audience. It goes to quite a different audience, and therefore the role we might play would be to somehow or other get it—or something we would extract from it—into the hands of the industrial community.

We also had the suggestion of a directory of opportunities for internship sabbaticals and theses. How do you get people to identify what such opportunities are? Would you take a candidate to a particular company and ask if

this idea? Or this thesis? Would companies be willing to say, "Come to us, all who meet the following criteria?" Won't it be difficult to get such a statement out of any company officially?

Comment [Hopper]:
We had to do that with some of our local colleges to tell them that we *were* having some internships. We specifically interfaced with local colleges because we think that's more appropriate.

Comment [Glaser]:
As a practical matter, it's got to be a local matter. If Paul has an opportunity in Rochester or Stamford, could someone in San Jose take advantage of it?

Comment [Higgins]:
It seems to be local because schools must specialize in different things. Industry might specialize in something and the only school interested might be across the country. If you've got a committee that finds out who the points of contact are and submit them to dozens of industry organizations and ask them, you'd be surprised how seldom you get a specific proposal instead of a generalized one, addressed to the right guy in a letter form. W[illegible] your organization make spaces available? Here's the program. If you've gotten a Federal blessing on it already, here's the citation that makes it legal, here's the pact ruling, the prize. You're going to get some positive responses.

Comment:
How do we get that Federal blessing, Bill?

Comment [Higgins]:
Office of Management and Budget and Mr. Puckorius sitting next to you.

Comment [Glaser]:
Mr. Puckorius, how do we get such a blessing?

Comment [Puckorius]:
I'd be happy to help but there isn't any funding available. There are some authorizations buried in the fine print in the government and I think with OMB's help, we can find some of them. We'd have to go with any existing funding for personnel.

Comment:
We might find out from AACSB how the Sears program works. There's no reason why you couldn't have something very similar to it. A similar program with government and industry should be possible.

Comment:
The DPMA chapter structure might be a way to handle this nationally and informally.

Comment:
Yes, there is that structure, and other professional society structures that are akin to it.

Comment [Glaser]:
Can anyone comment on the graduate assistantship directory? Does it work? Is it possible to identify on a timely basis what openings there are, to publish them, and to get them into the hands of prospective graduate assistants in time for them to respond within the annual cycle of such processing?

Comment:
It's possible, but the directory is sometimes out of date; for example, you may have had 10 positions available when it went out two years ago and now you've only got three.

Comment:
I'd worry about the lead time on such directories.

Comment [Glaser]:
I'm troubled a bit by the idea of publishing a directory. By the time you get all the mistakes out of it, publish it, mail it and people react to it, is it of any value?

Paul Strassmann's group suggested a directory of available industry personnel to teach courses. Again, isn't that likely to be a highly local matter? Could an individual take on a continuing responsibility at a school which was very far removed from his base of operations?

Comment
No, but presumably an individual could put some geographical constraints on availability.

Comment:
I think Paul's group had in mind not only teaching a short course or even a semester, but also a seminar. Individuals could announce a topic area they were prepared to speak about and then would negotiate with the school for the time—like a speaker's bureau.

Comment [McKenney]:
It seems to me that in the case of both the directory of opportunities and speakers you're making two kinds of markets. In one sense you're being a broker, but in the other sense you're announcing to the world at large that industry is, in fact, interested in this activity, that it's important and that people are willing to spend time and energy on it. Everyone is used to having out-of-date directories. The fact

that there's a list of things that was at one time relevant may make an impact on others. If this campus is putting in such a seminar, why not you?

Comment:

One point that has come through loud and clear today from so many of the things that have been said is that a number of you feel you became aware of activities that you would not otherwise have known about and that there's very little communications other than the "bamboo telegraph" to find out what other people are doing. You call Harlan, and he refers you to Jim McKenney, and Jim refers you to someone else, and you wind up talking to someone in London. But there really is no other mechanism that's very useful for getting this information around. And Jim, you make a very good point—what you're trying to do is to create an atmosphere. One way to create an atmosphere is to make everybody who isn't in one of these programs feel bad about it because he's not doing his share. Or, to put it in more crass terms, that he's not benefitting from it as perhaps others are, even his competition. People should feel left out if they aren't part of such a program.

Comment [Kapur]:

For me, the problem is getting people to come and speak on different subjects of their expertise. I've had to call maybe 30 persons to get the right one. A directory of speakers would even help the teachers very much—they could invite persons who had an intention of being available to talk on certain subjects.

Comment [Glaser]:

Speakers bureaus are not cheap to administer. ACM has had a very successful lectureship program for years. But, with only 40 or 50 lecturers it costs $8-10,000 a year to administer that program, including travel money.

The role of the Foundation as catalyst came up several times and indeed that's a role we'd like very much to play. In addition to DPMA, we have access to AFIPS, ACM, ICCP and several others. In fact, our charter is so broad that we can do whatever we have to do to get things done.

So, we would like to play the catalytic role, wherever possible ignoring political lines that are already drawn, because these usually are not very helpful. Mike Samek asked me for the opportunity to make a comment on that subject.

Comment [Samek]:

I think this is about as good an audience as I can possibly find to suggest something for which I have no solutions, but I can certainly point out the problem.

Having been involved at one time or another in professional societies, and numerous other meetings like this, I'd like to make a plea that we do something about the fragmentation we now have. I'd like to make the subject "Defragmentation."

If you take somebody like me out of industry, who is trying to participate as much as he can in such meetings, it's almost impossible. I could spend a tremendous amount of time in very useful meetings, but I simply can't afford the time to do it.

There are simply not enough people resources to go around (not to speak of money) to really support so very many different efforts.

I think it also doesn't help us to be a spokesman for only some part of this industry of discipline or profession (which we really don't have and which I very much think we ought to have). I don't necessarily mean just political clout—I'm not looking for that—but something that more likely expresses the concensus of a fairly large profession.

I don't believe that there's any need to concern oneself about the need for diversity or differences of outlook, or interests or opinions. That could very well be accommodated in some less-fragmented way than we have now. ACM is doing it quite successfully with SIGs and SICs and other such specialized groups.

But I find it very difficult, really, to somehow make a contribution on the basis on which it now has to be done. You're talking about the Education Foundation here; ACM is talking about something else, etc. The decision of where you go becomes strictly one of official interrelationships, and I just don't think that's right.

Let me come to the even more telling part of this fragmentation (and that's) the question of money. The support, financial as well as moral I think, that we get as a profession is absolutely "nil," and that includes, by the way, in my opinion, educational institutions. But, having been somewhat involved in the problem at ACM and more recently in what SMIS does, I think there are a total of about 40 corporations that to some small degree support the professionalism of what we do. If you consider the

pervasiveness of our discipline in industry, government and educational institutions, and if you consider the huge budgets that we deal with, I think that on a percentage basis we are absolutely nowhere. Something's got to be done. If we keep on differentiating, it's only going to get worse.

I have no solution, but I would like to point out that other technical organizations have been able to overcome this.

Comment [Higgins]:

Most every data processing operation I know of is spending a fantastic amount of money and time trying to overcome the lack of education. If we could only somehow get industry to recognize the cost of fumbling around, then the amounts you're talking about wouldn't look very large.

There's a cartoon that says, "We never have time to do this thing right the first time, but we always have time to do it twice." We educate people through hard knocks but we don't call it education. We call it the cost of doing business. I agree that the stakes are fantastic.

Comment:

Maybe we should be thinking of forming committees in these areas to define problems and to look for somebody else to do the job of authorship. Come up with something that will help a company make an honest buck, reduce costs, do the job better. They will pay for copies, and the author and your association could share the royalties. The acid test of the marketplace would determine whether you survive or not.

Comment:

I basically support the recommendations that Paul Strassmann made in his group: These programs look like something that would defragment things, if only we can do something about them.

Comment:

I don't know anyone who doesn't agree that there are too many professional groups. But, fortunately or unfortunately, we suffer from our own prosperity these days. Fifteen or 20 such groups are all extremely prosperous and doing what most people agree is a good thing; they just are not very interested in solving the problem that Mike mentioned. And I think the only way they are going to get very interested in solving that problem, is a loss of prosperity. That would cause them to band together and work for survival. I just don't see that happening in any of the foreseeable future; several of them were in difficulties a few years ago, but most of them are out of it now, and seem very prosperous.

Comment [Glaser]:

Dick Canning's group suggested that we make contact with company foundations, presumably, the ones represented by people in this room and others. And also, Dick, you used the expression "endorsement by industry." It seems to me that the acid test of endorsement is funding. We will be asking you to put us in contact with the foundations in your company; and we may also ask for an introduction. Perhaps you would be good enough to accompany us when we have a specific proposal to discuss with your foundations. We obviously can't ask someone to fund us just because we are good guys. We have to have good ideas.

One last item, Dallas Wolf's group suggested more publishing by industry people of useful work they have done. How do we get people in industry to publish the results of their work and to share it with others?

Comment [Walker]:

We do publish insurance type information but, as far as we are concerned, you people aren't interested in reading it. Sears, too, does their thing but I don't have much interest in what Sears is doing. I do, however, have a decided interest in what Allstate is doing.

Comment:

Organizations like the American Petroleum Institute, American Bankers Association and Grocery Manufacturers Association, for example, sponsor operations and automation conferences. Somebody could try to get copies of the proceedings and ask permission to reprint, or at least publish a critical bibliography.

The information is there, but we're all reading our own thing and not finding out what other people are doing in related areas.

Comment [Glaser]:

Jim McKenney, if you wanted material for a course on what's happening in banking, where would you go to get it? Is anything published which would describe this year's banking "breakthroughs?"

Comment [McKenney]:

I've been an editor of the Harvard Business Review for about eight years. People like to tell success stories; but when there are two successes, the second story makes very dull reading and makes you either feel envious or guilty. Getting stories

of painful learning experiences is nearly impossible. We have tried to capture the trials and tribulations of three large insurance companies in cleaning up data bases only to be in there for six months and then be gently asked (as they fire the manager), "We would appreciate it if you wouldn't take the time to write us up." But we need information like that and I don't know how to get is.

I think the best publication we have is Canning's. Maybe we need a couple more of him.

Comment [Glaser]:
There were two recommendations from Felix's group: One to codify what the field is, and two to evaluate the user's needs. The Foundation recently conducted a survey of DPMA members on educational needs and Gene Smith will give you just a few highlights of the results. They soon will be published in more detail in *Data Management*.

Comment [Smith]:
This was grass-roots survey of the educational needs and desires of EDP professionals. We sent out about 21,500 questionnaires and got an 18 percent response (3,884 to date). Here are some of the results: 3,144 were interested in self-assessment to 3,498 felt that they needed further training. Of these, 3,294 wanted training in DP. Of these, 2,774 training in areas such as DP management assessment (which ranked third), verbal communication, written communication, memory training, speed reading and problem solving.

We asked people to distinguish between their educational needs and those of the people they supervise. For the people they supervise "verbal and written skills" ranked highest, bearing out some of our earlier discussions. "Teach them to read and write and we'll teach them Fortran."

Comment [Glaser]:
In the five minutes that we have left, tell us what do you want the Foundation to do? Given all these opportunities that you've laid out for us, what do you think we ought to do first?

Comment:
Circulate the discussion of this meeting. I think that would be a very good way to stimulate cooperation.

Comment:
Hopefully a good news release will be coming out soon after this meeting.

Comment [Glaser]:
We hope so. *Datamation* is standing by for a story. (Ed. note: Datamation, May 1976, pg. 198). You can expect a request to join a steering committee for one or more projects.

Once we have sorted out the many ideas that you've given us today, we're going to call on you even if you don't volunteer, because we need all the help we can get.

Appendix

BARNES, Dr. B.L.
Dean, College of Business
University of Iowa
Iowa City, IA 52240

BELLAN, Theodore M.
Vice President, Computer Services
McDonnell Douglas Automation Company
P.O. Box 516
St. Louis, MO 63166

BOWEN, Charles R.
Director, Plans & Program Administration
IBM Corporation
Old Orchard Road
Armonk, NY 10504

BRYANT, Dr. Howard
Chairman & Associate Professor of Information
School of Business Systems
Virginia Commonwealth University
901 W. Franklin St.
Richmond, VA 23284

BUCKLEY, William P.
Systems Supervisor
Aluminum Company of America
1501 Alcoa Building
Pittsburg, PA 15219

CANNING, Richard G.
Canning Publications, Inc.
925 Anza Avenue
Vista, CA 92083

CURRY, Bruce G.
Vice President, Management Systems & Services
Hertz Corporation
660 Madison Avenue
New York, NY 10021

HIGGINS, Colonel William W.
Director for Data Automation
Office of Assistant Secretary of Defense (Comptroller)
Washington, D.C. 20301

HODGES, J. Les
General Manager, Computer Services
Texaco, Inc.
P.O. Box 52332
Houston, TX 77052

HOPPER, Max D.
Assistant Vice President - SABRE
DP & Communication Service
American Airlines
3800 N. Mingo Road
Tulsa, OK 74151

KAPUR, GOpal
Kapur & Associates
776 El Cerro Blvd.
Danville, CA 94526

KASKEY, Dr. Gil
Director
Sperry Univac Education Center
P.O. Box 1110
Princeton, NJ 08540

KAUFMAN, Felix
National Director, Management Consulting Services
Coopers & Lybrand
1251 Ave. of the Americas
New York, NY 10020

LEATHERMAN, J. Ralph
President, DPMA
Hughes Tool Corporation
P.O. Box 2539
Houston, TX 77001

OTT, Robert H.
National D-P-Mgr.
Sears Roebuck Company
Sears Tower
Chicago, IL 60684

MILLS, Dr. Harlan
IBM
Federal Systems Division
18100 Fredrick Pike
Gaithersburg, MD 20760

MORGAN, Professor Howard
The Wharton School
University of Pennsylvania
Philadelphia, PA 19174

McKENNEY, Professor James
Harvard University
Graduate School of Business
Boston, MA 02163

PAIR, Paul M.
Senior Education Consultant to Division Gen. Mgr.
Control Data Institute
430 N. Michigan Ave.
Chicago, IL 60611

PUCKORIUS, Honorable Theodore D.
Commissioner, Automated Data & Telecommunications Service
General Services Administration
Washington, D.C. 20405

RALSTON, Dr. Anthony J.
President, AFIPS
SUNY at Buffalo
4226 Ridge Lea Road
Amherst, NY 14226

ROARK, Mayford L.
Executive Director, Systems Office
Ford Motor Company
The American Road
Dearborn, MI 48121

SAMEK, Michael J.
Vice President, Management Services
Celanese Corporation 63
1211 Ave. of Americas
New York, NY 10036

SIMPSON, Warren
Director, Regional Training Center
U.S. Civil Service Commission
230 South Dearborn
Chicago, IL 60604

SMITH, Charles W.
Exxon Corporation
P.O. Box 153
Florham Park, NJ 07921

SMITH, Jr., Robert L.
Asst. to Exec. Vice President
Office of Exec. V.P. for Admin.
TAMU SYSTEM
College Station, TX 77843

STRASSMANN, Paul A.
Director, ADM & Info. Systems & General Mgr., Info Serv. Division
Xerox Corporation
Stamford, CT 06904

TEICHROEW, Professor Daniel
Chairman, Dept. of Industrial Engr.
University of Michigan
231 W. Engineering
Ann Arbor, MI 48104

WALKER, Merton R., CDP
Assistant Vice President, Mgt. Info. Serv.
State Farm Fire & Casualty Co.
112 E. Washington Street
Bloomington, IL 61701

WILSON, Dr. Howard B.
General Director, Mgmt. Information Systems
Reynolds Metals Corporation
6603 W. Broad St.
Richmond, VA 23261

WITHINGTON, Frederic G.
Senior Consultant
Arthur D. Little, Inc.
Acorn Park
Cambridge, MA 02138

WOLF, D.R.
General Manager, Information & Computer Service
Shell Oil Company
P.O. Box 20329
Houston, TX 77025

Board of Regents

T.M.

Data Processing Management Association Education Foundation

Dedicated to expanding educational opportunities for information systems professionals

President

Roland D. Spaniol, Ph. D., CDP, Director of Computer Services, Professor of Management, Eastern Illinois University, Charleston, Ill.

Vice President

Eugene B. Smith, Ph. D., CDP, Associate Professor, Department of Analysis and Research, Texas A & M University, College Station, Tex.

Regent

James A. Campise, P.E., CDP, Consultant and faculty member of the Department of Mathematical Science, Rice University, Houston, Tex.

Regent

George R. Eggert, Chief, Office of Planning and Management, DCASR, U.S. Department of Defense, Chicago, Ill.

Regent

George Glaser, Consultant, San Mateo, Calif., former President of American Federation of Information Processing Societies.

Executive Director and Secretary-Treasurer

T. David McFarland, CAE, Executive Director, Data Processing Management Association, Park Ridge, Ill.

T.M.

Data Processing Management Association Education Foundation
505 Busse Highway, Park Ridge, IL 60068
312/825-8124

69

research/programs/services:

for the information systems professionals, the industry, educators, business, government, and the public

The Data Processing Management Association Education Foundation, a not-for-profit organization, was established to meet the changing requirements of the highly dynamic information processing profession and industry.

Because of the complexity of computer technology, and the need to develop effective professional personnel to coordinate, supervise, and perform continually expanding and diversified operations, it is imperative that sound, comprehensive, professional-level educational programs be created, maintained and promulgated.

The Education Foundation, as an independent body representing practitioners, educators and researchers, seeks to provide leadership in identifying educational opportunities that will advance the information systems profession.

An immediate Foundation objective is the dissemination of information among technical societies and the entire information systems community. Under consideration are such projects as a survey to determine the educational needs and desires of professionals; a conference to bring industry and university leaders together to examine the current status of business data processing education and to improve the interface between the two communities; and to determine specific needed educational materials.

The Foundation is governed by a five-member Board of Regents who are appointed by the current International President of DPMA to serve overlapping three year terms. Four Regents must be information systems professionals, educators or researchers and the remaining member may be appointed without regard to profession. Three of the Regents are selected from among the membership of DPMA and the remaining two without regard to such membership.

Grants are sought to conduct special studies. Results of these studies will be published as monographs, magazine articles, or as special reports.

Any grant will be tax exempt since the DPMA Education Foundation is exempt from Federal income tax under section 501(c)(3) of the Internal Revenue Code.

Objectives of the Foundation:

- To advance the education of information systems professionals, the industry, educators, business, government, and the public.
- To determine the needs and desires of information systems professionals with regard to education and development.
- To conduct research and development activities related to educational courses and programs.
- To develop and stimulate the use of programs for self-assessment.
- To develop and stimulate the use of educational programs for information systems evaluation.
- To further the development of effective information systems curricula.
- To encourage the development and adoption of technical and managerial information systems standards and preferred practices.
- To develop and release publications in support of the Foundation's objectives.
- To provide funding sufficient to attain these objectives.